CS-566 Deep Reinforcement Learning

Introduction

Nazar Khan
Department of Computer Science
University of the Punjab

What is Deep Reinforcement Learning?

- Deep RL studies how to solve **complex problems** that require making a **sequence of good decisions**.
- These problems often live in **high-dimensional state spaces**:
  - Many variables must be considered simultaneously.
  - Example: In chess, the position of each piece defines the state; there are more possible states than atoms in the universe.
  - Example: In robotics, sensors may produce hundreds or thousands of readings per time step.

Examples of Sequential Decision-Making

- **Making Tea:** wait until water is boiling, add tea leaves, adjust milk, control sweetness, simmer for flavor, strain before serving.
- **Tic-Tac-Toe:** sequences of moves, opponent's responses, and planning ahead.
- **Chess:** much more complex version of tic-tac-toe with astronomical state space.
- **Having a Conversation:** listen to the other person, interpret context, choose a relevant response, maintain flow, achieve an agenda.

Success comes from a **sequence of decisions**, not a single one. Each decision has an immediate consequence and a long-term consequence. An RL agent learns through trial-and-error.

State Spaces

X		
	O	
O		X

Tic-Tac-Toe

$3^9 = 19{,}683$ possible boards.

Chess

$\approx 10^{47}$ possible states.

Go

$\approx 10^{170}$ possible states.

Conversation

Infinite possible states.

What is Deep Reinforcement Learning?

- Combination of **deep learning** + **reinforcement learning**
- Goal: learn optimal actions that maximize reward across all states
- Works in high-dimensional, interactive environments

Deep Learning

- Function approximation in high dimensions
- Uses deep neural networks
- Examples: speech recognition, image classification

Reinforcement Learning

- Learns from trial and error, not from fixed datasets
- Feedback comes from the environment (reward / punishment)
- Builds a **policy**: which action to take in each state

Where DRL Fits

	Low-Dimensional	**High-Dimensional**
Static Dataset	Supervised Learning	Deep Supervised Learning
Interaction	Tabular RL	*Deep RL*

Tabular RL

State	Action
State 1	Move Left
State 2	Move Down
⋮	⋮
State N	Move Right

Deep RL

$$f(\text{state}) \rightarrow \text{action}$$

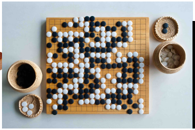
$$f(\text{state}) \rightarrow p(\text{action})$$

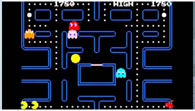
$$f(\text{state}) \rightarrow \text{probability distribution}$$

Applications of DRL

- Robotics: locomotion, manipulation, pancake flipping, helicopters
- Games: Chess, Go, Pac-Man, StarCraft
- Real-world: healthcare, finance, recommender systems, energy grids, ChatGPT

Picture: Max Ewen Central Model Helicopter Club





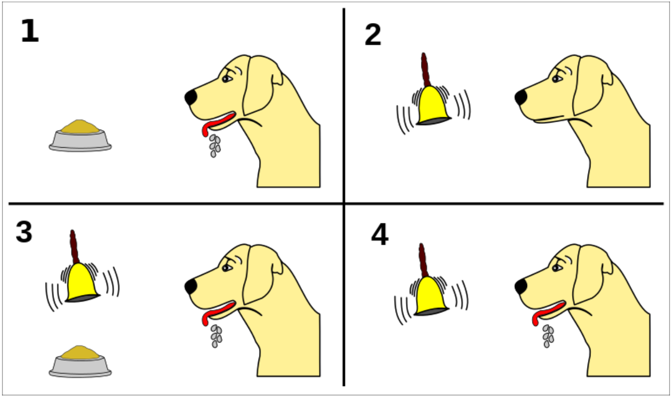
Four Related Fields

1. Psychology

- Conditioning: Pavlov's dog
- Operant conditioning (Skinner)
- Learning from reinforcement is a core AI idea

Four Related Fields

1. Psychology



Pavlov's dog: A natural reaction to food is that a dog salivates. By ringing a bell whenever the dog is given food, the dog learns to associate the sound with food, and after enough trials, the dog starts salivating as soon as it hears the bell, presumably in anticipation of the food, whether it is there or not.

Four Related Fields

2. Mathematics

- Markov Decision Processes (MDPs)
- Optimization, planning, graph theory
- Symbolic AI: search, reasoning, theorem proving

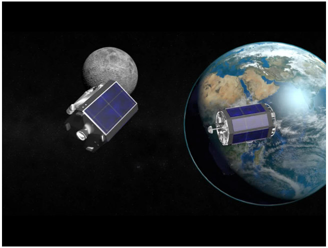
А. А. Марков (1886).

Andrei Markov (1856-1922)

Four Related Fields

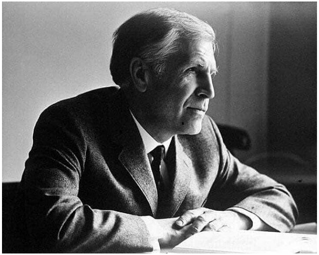
3. Engineering

- Known as **optimal control** in engineering.
- Focus on **dynamical systems**.
- Bellman and Pontryagin's work in optimal control laid the foundation of RL.



Two space vehicles docking

Richard Bellman (1920-1984)

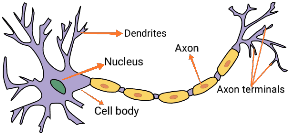


Lev Pontryagin (1908-1988)

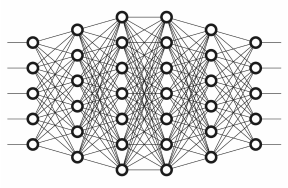
Four Related Fields

4. Biology

- Connectionism: swarm intelligence, neural networks
- Nature-inspired algorithms: ant colony, evolutionary algorithms



Biological Neuron



Artificial Neural Network

Hinton, LeCun, Bengio

Three Paradigms of Machine Learning

- Machine Learning studies how to approximate functions $f : X \to Y$ from data.
- Often, functions are not known analytically.
- Instead, we *learn* them from observations.
- Three main paradigms:
  1. Supervised Learning
  2. Unsupervised Learning
  3. Reinforcement Learning

Functions in AI

- A function transforms input x to output y: $f(x) \to y$.
- More generally: $f : X \to Y$, where X, Y can be discrete or continuous.
- Real-world functions may be stochastic: $f : X \to p(Y)$.

Given vs. Learned Functions

- Sometimes f is given exactly (laws of physics, explicit algorithms).

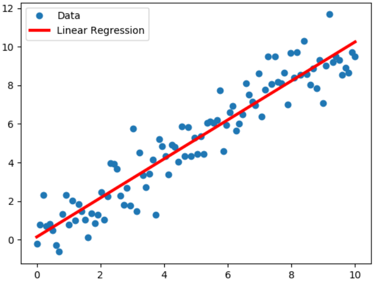
Example: Newton's 2nd Law $F = m \cdot a$.

- Often, f is unknown and must be approximated from data.
- This is the domain of **machine learning**.

Supervised Learning

- Data: example pairs (x, y).
- Goal: learn a function $\hat{f}$ that predicts y from x.
- Common tasks:
    - Regression: predict a continuous value.
    - Classification: predict a discrete category.
- Loss function measures prediction error, e.g. MSE or cross-entropy.

Regression



Classification

Cat

Cat

Dog

Cat

Dog

Dog

Unsupervised Learning

- No labels: only input data x.
- Goal: find structure in data (clusters, latent variables).
- Examples:
  - k-means clustering
  - Principal Component Analysis (PCA)
  - Autoencoders
- Learns $p(x)$ instead of $p(y|x)$.

Reinforcement Learning

- Third paradigm of machine learning
- Learns by **interaction** with the environment
- Data comes sequentially (one state at a time)
- Objective: learn a **policy** – a function mapping states to the best actions

$$\pi(s) \rightarrow a$$

Agent and Environment

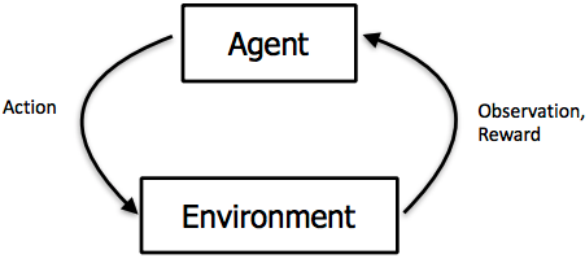


Figure: Agent interacts with Environment to maximize reward.

- **Agent:** Learner/decision-maker
- **Environment:** Provides feedback and state transitions
- **Goal:** maximize long-term accumulated reward

Key Differences from Supervised/Unsupervised Learning

1. **Interaction-based:** No pre-collected dataset; data generated dynamically via interaction between agent and environment
2. **Reward signal:** Partial numeric feedback, not full labels
    - UL has no labels
    - RL has reward
    - SL has complete labels
3. **Sequential decision-making:** Learns policies across multiple steps

Key Differences from Supervised/Unsupervised Learning

- In RL there is no teacher or supervisor, and there is no static dataset.
- RL learns a policy for the environment by interacting with it and receiving rewards and punishments.

> - SL can classify a set of images for you.
> - UL can tell you which items belong together.
> - RL can tell you
>   - the winning *sequence* of moves in a game of chess, or
>   - the action-*sequence* that robot-legs need to take in order to walk.

Supervised vs Reinforcement Learning

	Supervised Learning	**Reinforcement Learning**
Inputs x	Full dataset	One state at a time
Labels y	Full (correct action)	Partial (numeric reward)

Implications of RL Paradigm

- Data is generated step-by-step ⇒ suited for sequential problems
- *Risk of circular feedback*
  - Policy selects an action and then learns from it.
  - If policy is poor, it will select a poor action.
  - Learning from a poor action might also be of low quality.
- RL can continue to learn indefinitely if environment is challenging
- Example: Chess, Go, robotics, conversational agents

Deep Reinforcement Learning

- Traditional RL: works on small, low-dimensional state spaces
- Many real-world problems: large, high-dimensional state spaces
- **Deep RL = RL + Deep Learning**
  - Handles large state spaces
  - Scales to complex tasks
- Key driver of recent breakthroughs in AI
  - Robotics `https://www.youtube.com/watch?v=I44_zbEwz_w`
  - Complex games like Chess and Go `https://www.youtube.com/watch?v=4PyWLgrt7YY`
  - Large Language Models (LLMs) `https://www.youtube.com/watch?v=WMmGzx-jWvs`

Summary

- Deep RL = deep learning + reinforcement learning
- Solves sequential decision problems in high dimensions
- Rooted in psychology, math, engineering, biology
- Applications: robotics, games, healthcare, finance, any interactive setting