NOVA IMS Information Management School

Reinforcement Learning

Lecture 1

Introduction to Reinforcement Learning

April 2025

Nuno Alpalhão
nalpalhao@novaims.unl.pt

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

UC Overview

Schedule

7 Lectures

- Every Tuesday at 8:30.
- Understanding the Basic Principles of RL.
- Variations of RL approaches and their implications.
- Real world applications.

7 Labs

- Every Wednesday at 8:30 (P3), Thursday** (P1) at 13:30 and Thursday** at 15:00 (P2).
- Hands-on experience of RL algorithms using Python.
- When to use RL?

** Classes on the 1st of May will be changed!!

Grading | First Season

1 Exam
(50%)

- 10 multiple choice questions and 5 other questions.
- 1:30 hours duration.

1 Project
(50%)

- Project will be the implementation of a RL application.
- Group Activity up to 4 people.
- To be delivered until the 20th of June.

Grading | Second Season

1 Exam
(50%)

- 1:30 hours duration.

1 Project
(50%)

- Project will be the implementation of a RL application.
- Individual.
- To be delivered until the 18th of July.

UC Overview

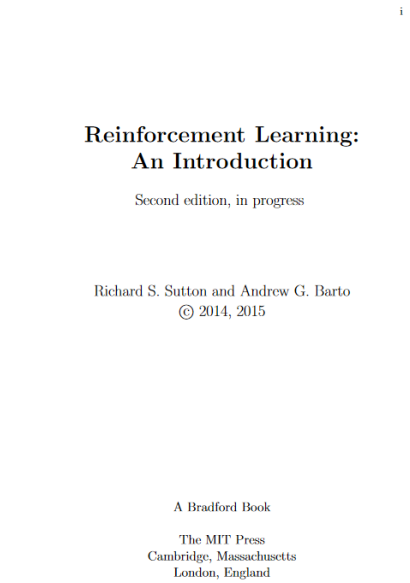
Tools

Programming Language

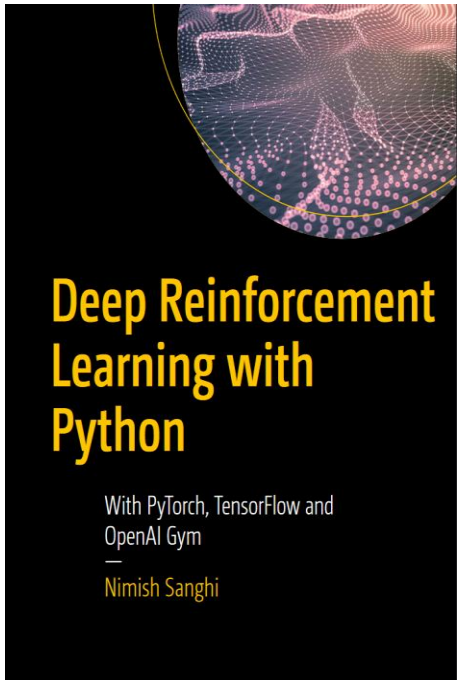
ML Libraries

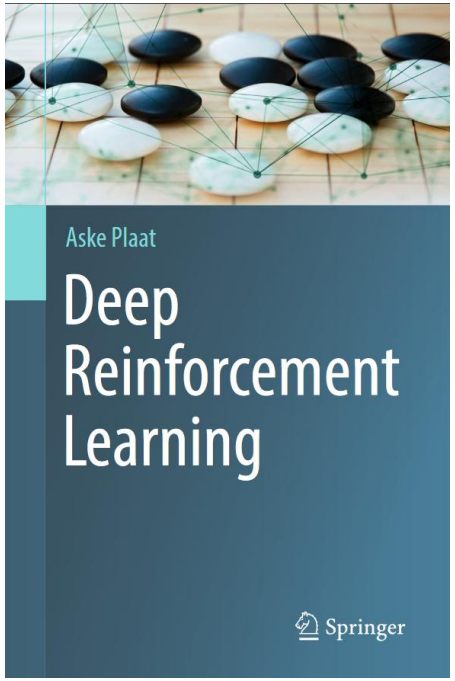
Environment Engine

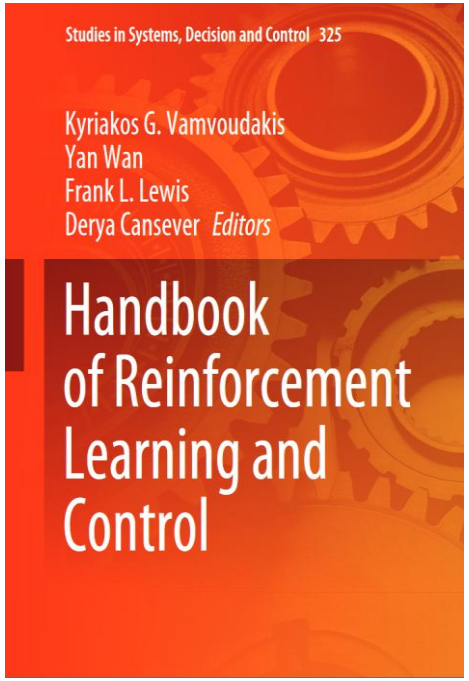
Bibliography



http://incompleteideas.net/book/the-book-2nd.html





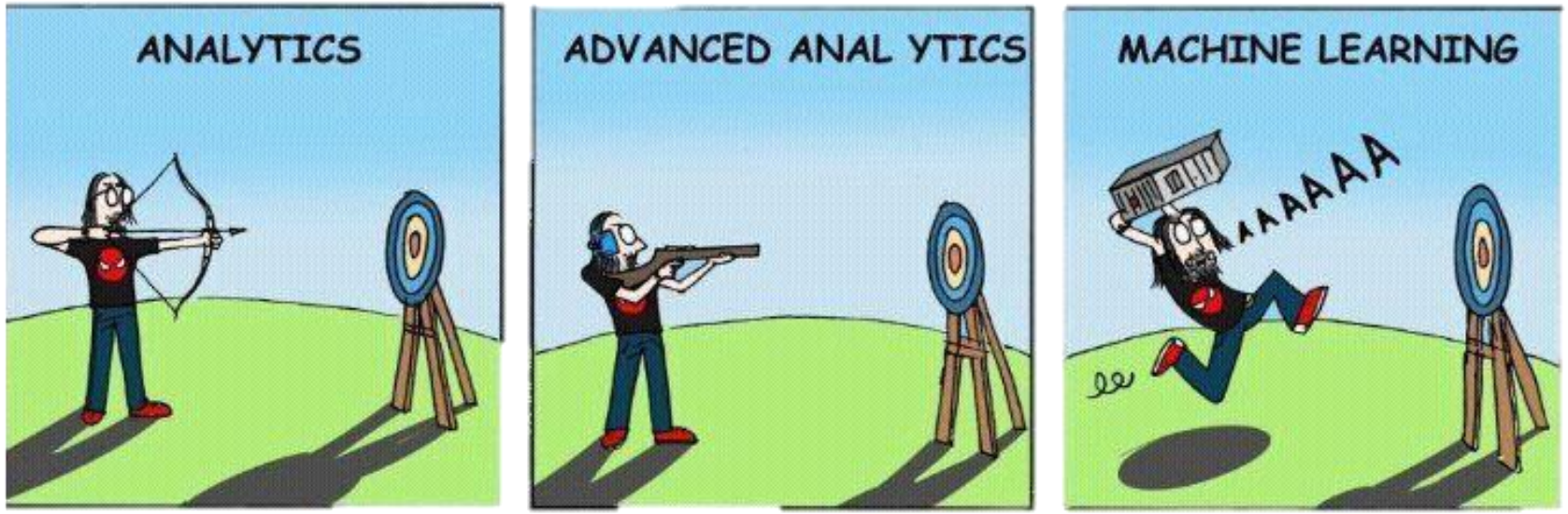


We will focus mostly on the first two books.

UC Overview

Learning Outcomes

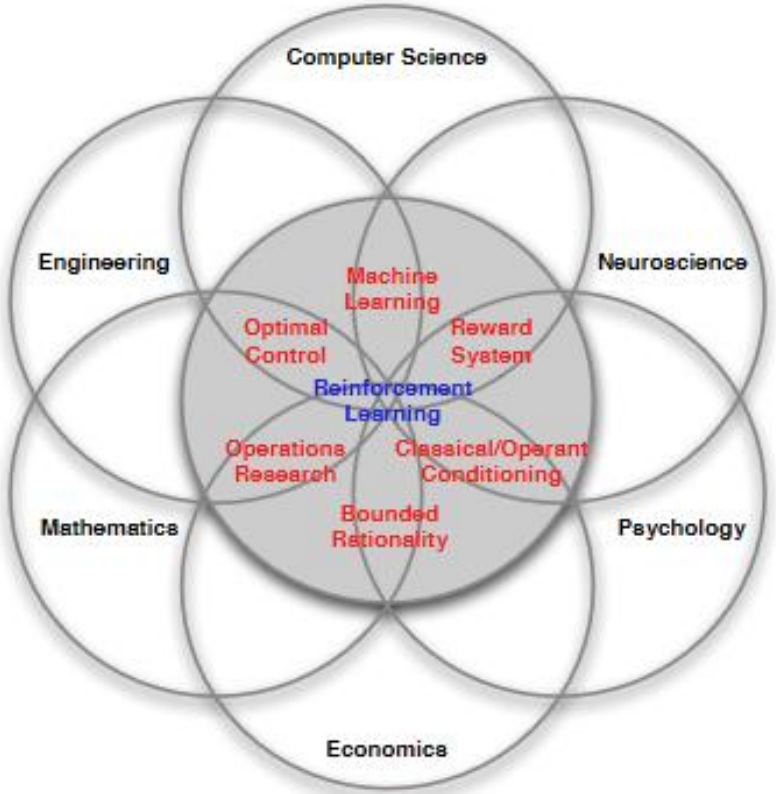
- Define the characteristics of reinforcement learning.
- Determine whether an application problem should be formulated as an RL problem.
- Formally define an RL problem (state space, action space, dynamics, and reward model).
- Be able to select which algorithm is best suited for a specific RL problem and justify it.

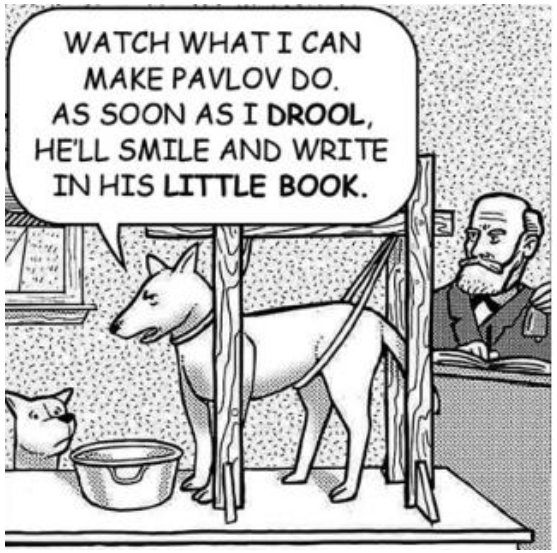


QUESTIONS?

Introductions to Reinforcement Learning

Many Faces of RL



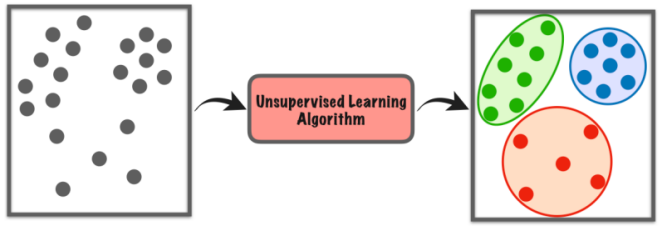


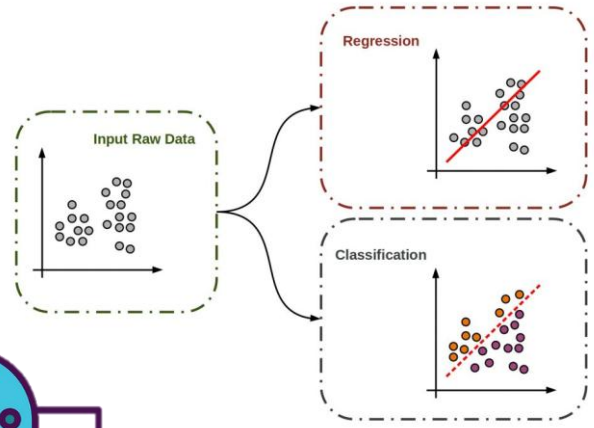
From DeepMind's and UCL with David Silver Course

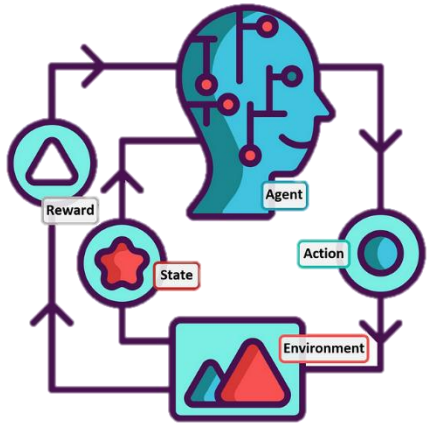
Introductions to Reinforcement Learning

Machine Learning

- Unsupervised Learning
- Supervised Learning
- Reinforcement Learning







Reinforcement Learning Involves

- Optimization
- Delayed consequences
- Exploration
- Generalization

Optimization

- Objective is to find an **optimal way to make decisions**.
- Yielding best outcomes or at least very good outcomes.
- Unequivocal notion of **usefulness of each decision.**

Delayed Consequences

- Decisions now have a long-term impact.
- Early rewards can **hinder the future**.

- Introduces two challenges
  - When **planning**: decisions involve reasoning about not just immediate benefit of a choice but also its long-term ramifications.
  - When **learning**: At each stage how can we **value the lessons learned**?

Exploration

- Learning about the world by making decisions.
- Agent as scientist.
- Only get a reward (label) for decision made.
- Decisions impact what we learn about. (What new information is obtained?)

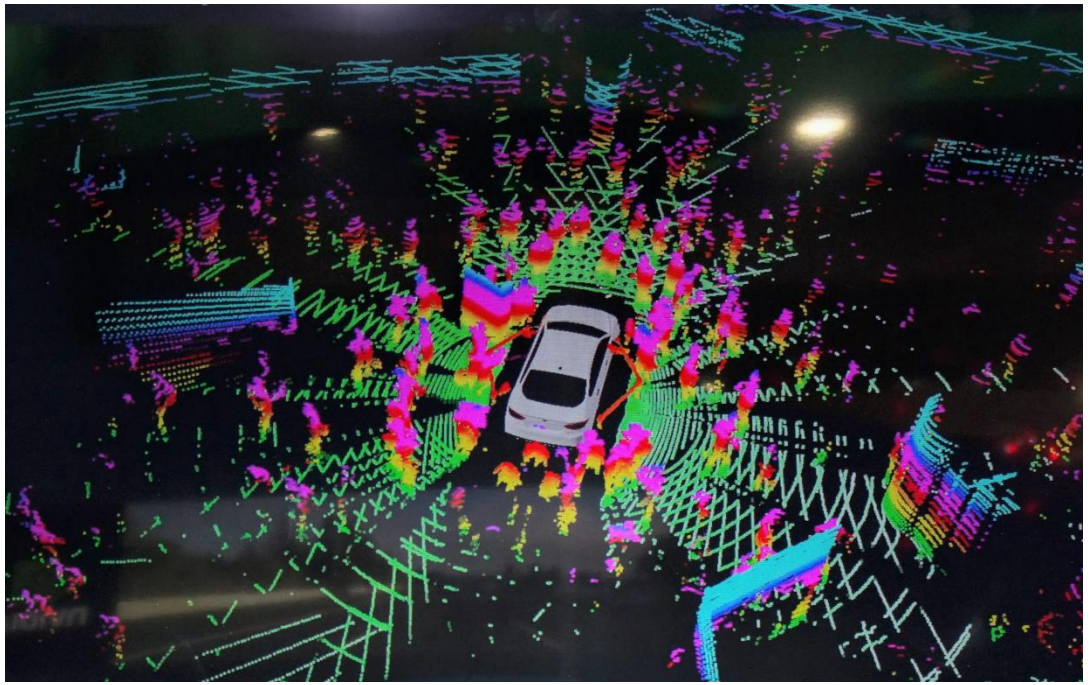
Generalization

- **Policy** is mapping from experience to action.

- Why not just pre-program a policy?

Autonomous Vehicles (AV's)

- AV's have sensors like LiDAR, radar, cameras, etc., that sense their nearby **environment**.
- The raw sensory data and object detection are combined to get a unified **scene representation** that is used for planning a path to a destination.
- The planned path is next used to feed inputs to the controls to make the **system/agent** follow that path.



Robots

- Using computer vision and natural language processing or speech recognition using deep learning techniques.

- Observe human behaviour through cameras and learn to perform like humans.

Thanks to machine-learning algorithms, the robot apocalypse was short-lived.

Recommendation Systems

- Video sharing/hosting applications YouTube, TikTok, and Facebook suggest to us the videos that we would like to watch based on our viewing history.

- Systems continually learn from the way users respond to the suggestions presented by the engine.

Introductions to Reinforcement Learning

Examples

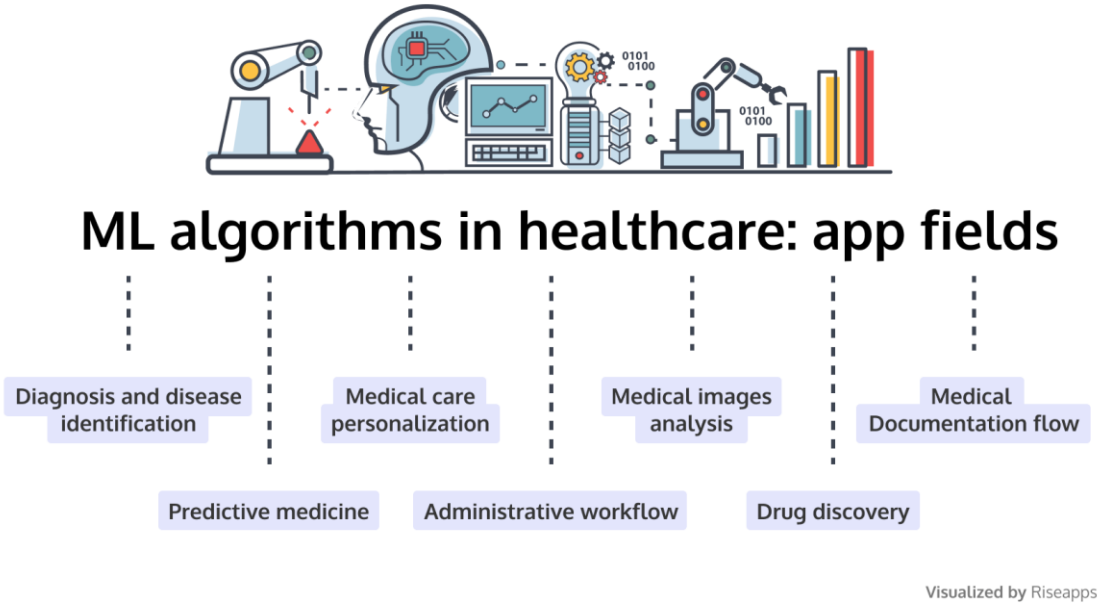
Finance and Trading

- Sequential action optimization focus wherein past states and actions influence the future outcomes.

- Reinforcement learning finds significant use in time-series analysis, especially in the field of finance and stock trading.

Healthcare

- Generating predictive signals and enabling medical intervention at early stages.
- Robot-assisted surgeries or managing the medical and patient data.
- RL-based systems provide recommendations learned from its experiences and continually evolve.



Reward Hypothesis

What makes reinforcement learning different from other machine learning paradigms?

- There is no supervisor, only a **reward signal**. For example, us humans can learn without examples of optimal behaviour.
- Feedback is delayed, not instantaneous.
- Time really matters (sequential, non i.i.d data), which means future interactions can depend on earlier ones.
- Agent's actions affect the subsequent data it receives.

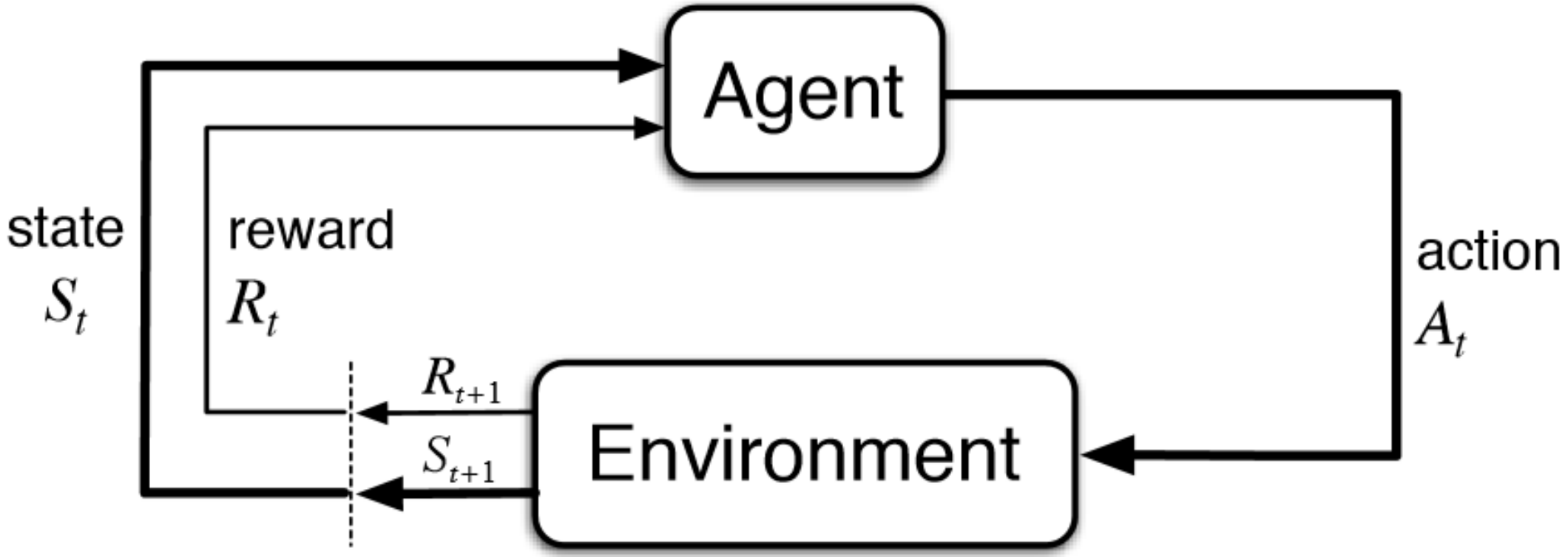
The constituents of deep reinforcement learning

	Low-dimensional states	High-dimensional states
Static dataset	Classic supervised learning	Deep supervised learning
Agent/environment interaction	Tabular reinforcement learning	*Deep reinforcement learning*

Supervised vs. reinforcement learning

Concept	Supervised learning	Reinforcement learning
Inputs x	Full dataset of states	Partial (one state at a time)
Labels y	Full (correct action)	Partial (numeric action reward)

RL Conceptualization

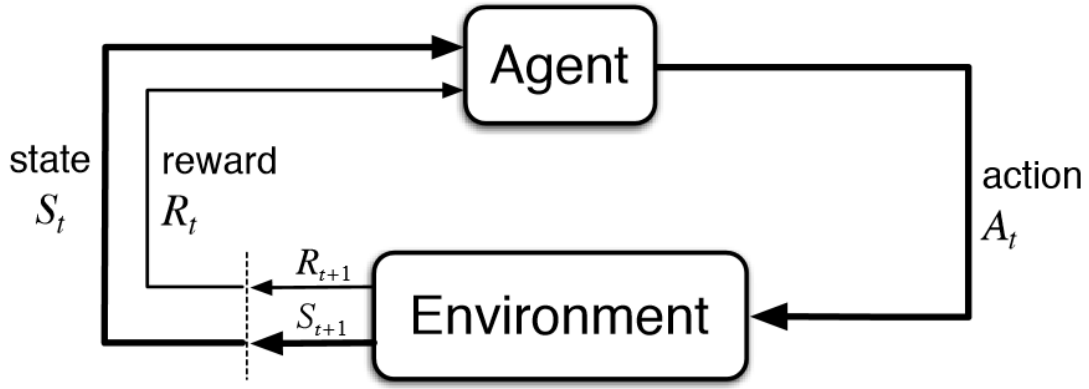


Reward Hypothesis

Reinforcement learning is based on the **reward hypothesis:**

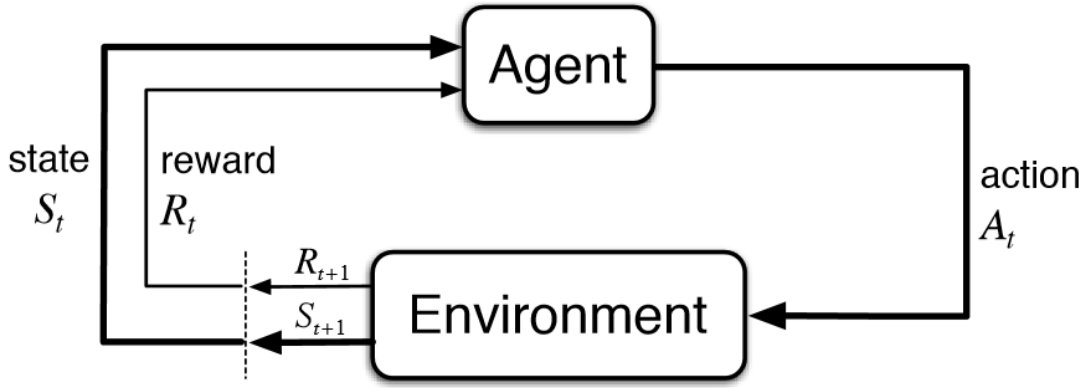
All goals can be described by the maximisation of expected cumulative reward.

- A reward is a scalar feedback signal.
- Indicates how well agent is doing at current step.
- The agent's job is to maximise cumulative reward.



The reinforcement learning formalism includes:

- Reward signal (specifies the goal).
- Environment (dynamics of the problem).
- Actions (What the agent can do).
- Agent:
  - Agent state.
  - Policy.
  - Value function.
  - Model.

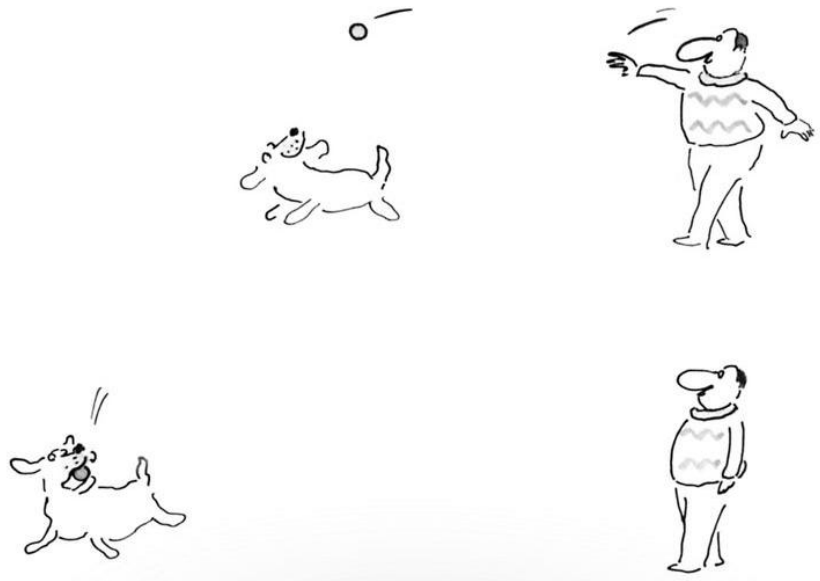


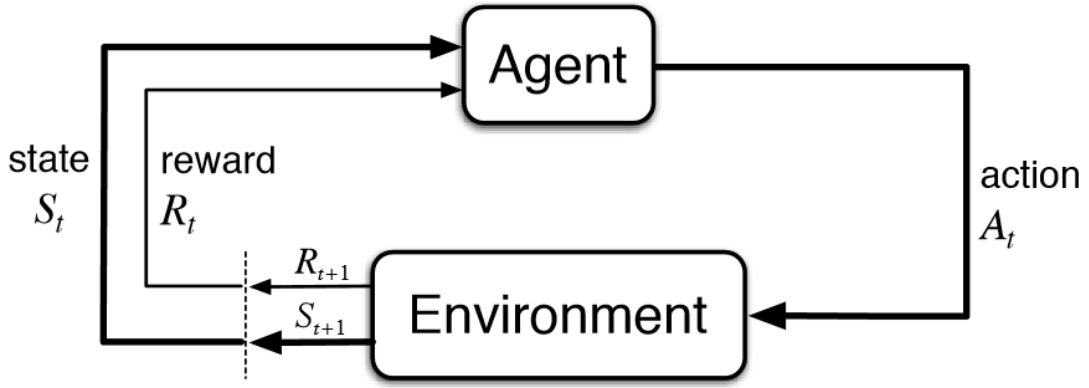
Introductions to Reinforcement Learning

Fetch I

Example: Let's play fetch!

- Reward signal: **Get a treat from owner.**
- Environment: **???**
- Actions: **???**
- Agent:
  - Agent state: **???**
  - Policy: **???**
  - Value function: **???**
  - Model: **???**



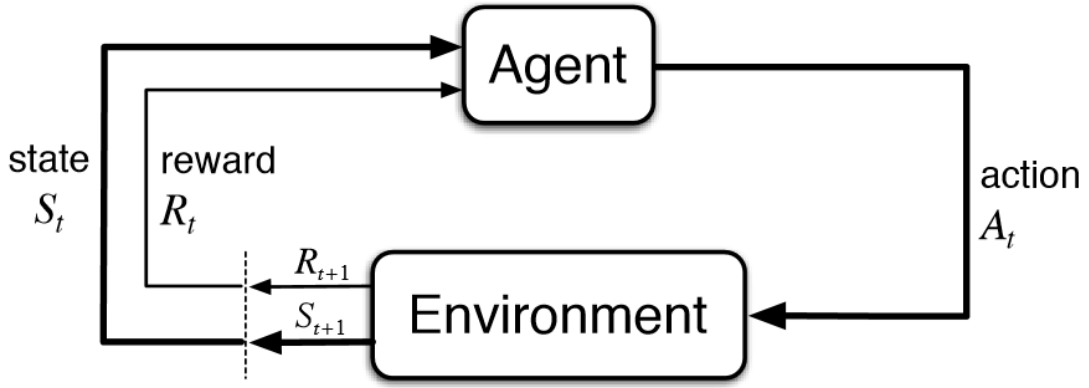


Environment State

- The environment state is the environment's internal state.

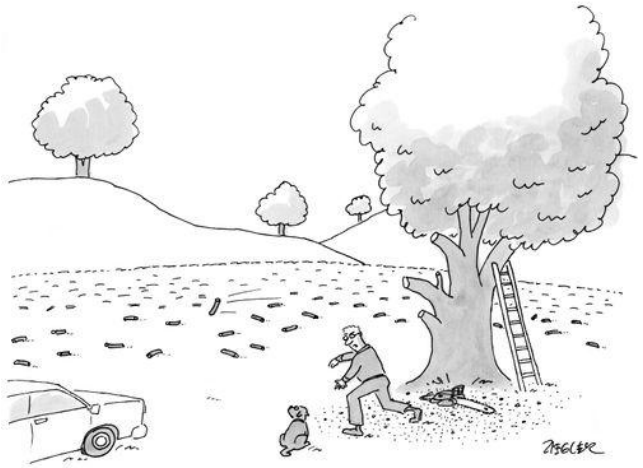
- It is usually invisible to the agent.
- Even if it is visible, it may contain lots of irrelevant information.

- Fully Observable Environments.
- Partially Observable Environments.

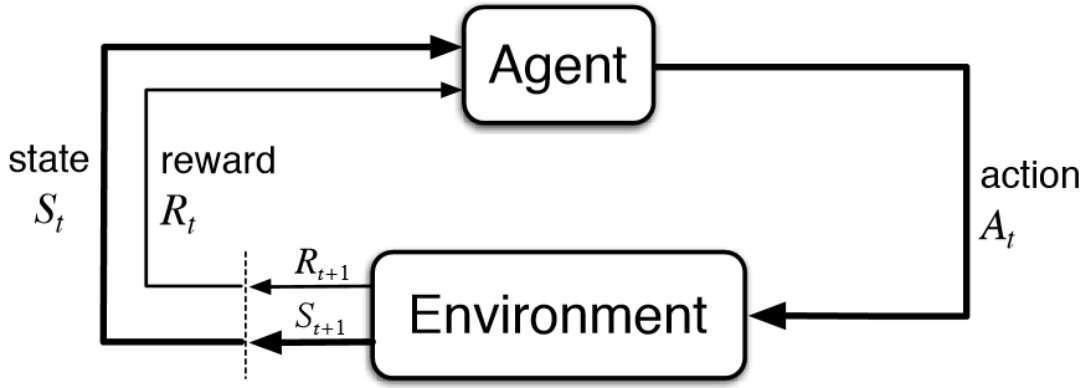


Example: Let's play fetch!

- Reward signal: **Get a treat from owner.**
- Environment: **Dog – Owner – Ball positions?**
- Actions: **???**
- Agent:
  - Agent state: **???**
  - Policy: **???**
  - Value function: **???**
  - Model: **???**

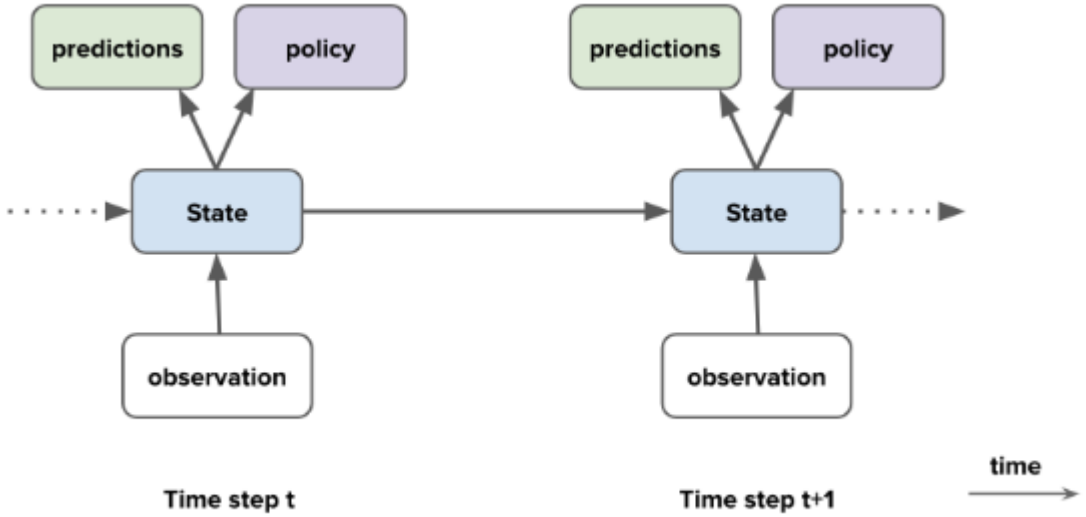


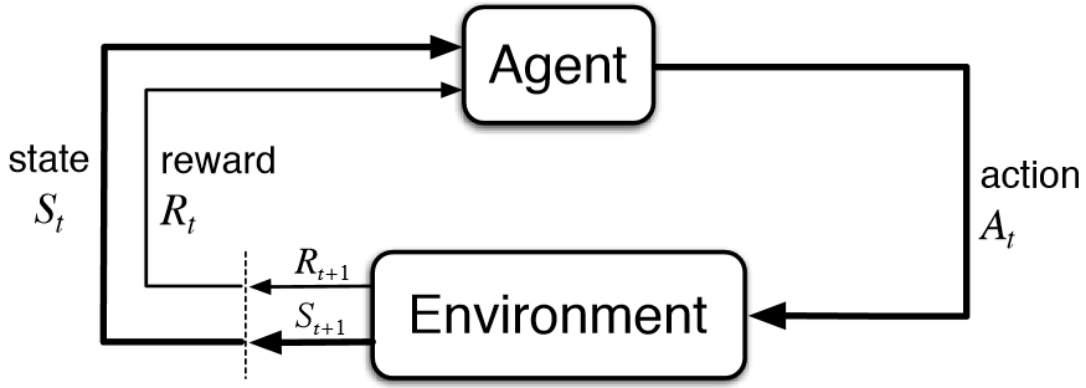
"How about, for God's sake, this one?"



Agent State

- The history is the full sequence of observations, actions, rewards.
- This history is used to construct the agent state.
- The agent's actions depend on its state.
- The **agent state is a function** of the history.
- The agent state is often much smaller than the environment state.



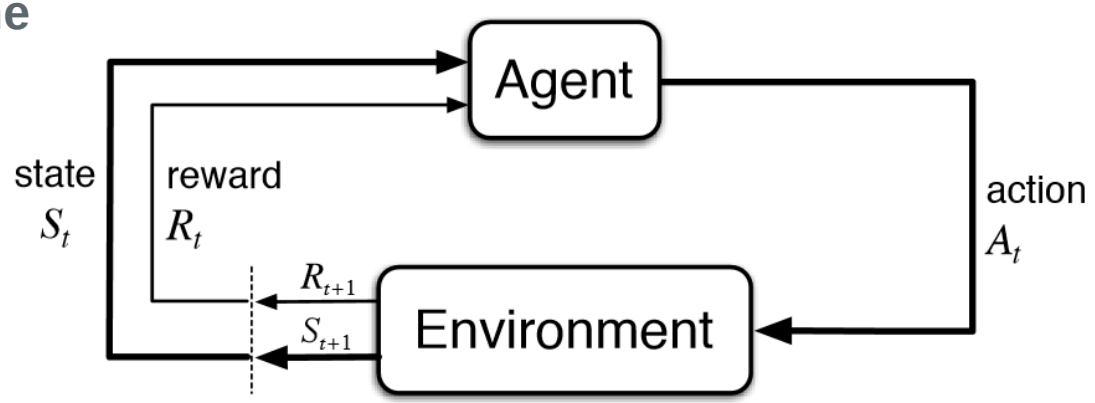


Introductions to Reinforcement Learning

Fetch III

Example: Let's play fetch!

- Reward signal: **Get a treat from owner.**
- Environment: **Dog – owner – stick positions?**
- Actions: **Move and move the stick.**
- Agent:
  - Agent state: **Current Position? Do I have the stick?**
  - Policy: **???**
  - Value function: **???**
  - Model: **???**

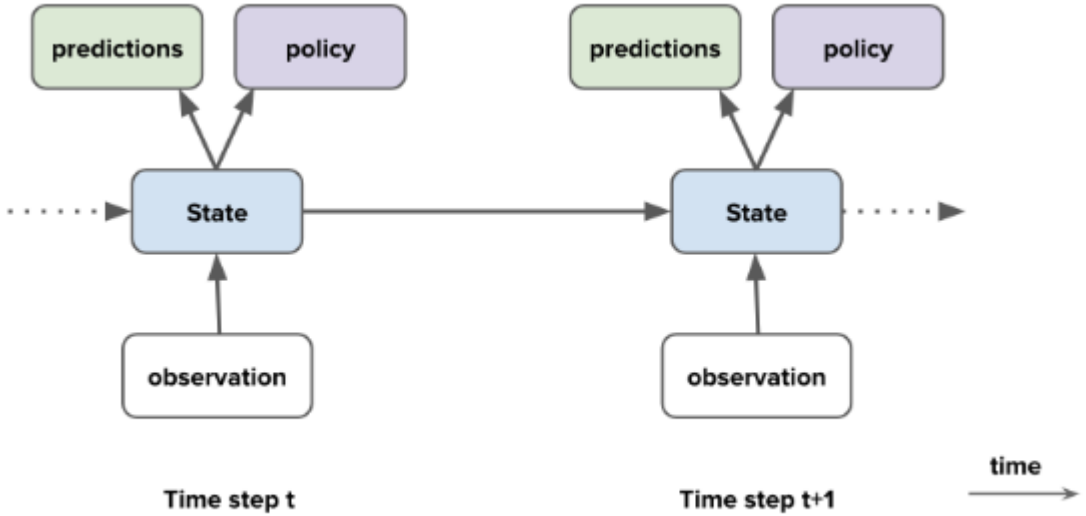


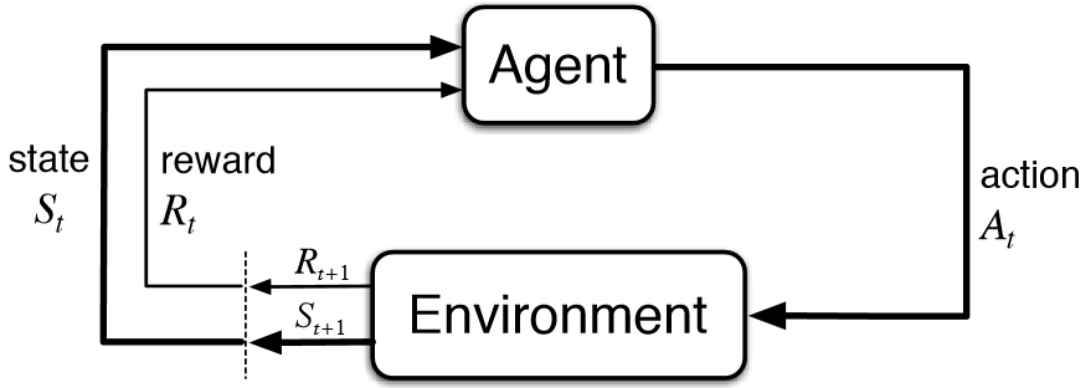
Agent Policy

Behaviour function

- A policy is the agent's behaviour.

- It is a map from state to action:
  - Deterministic policy.
  - Stochastic policy.



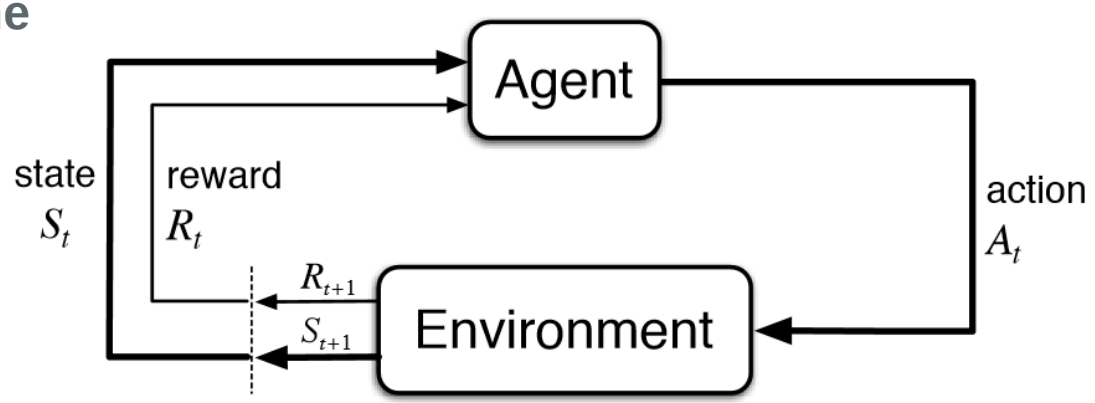


Introductions to Reinforcement Learning

Fetch IV

Example: Let's play fetch!

- Reward signal: **Get a treat from owner.**
- Environment: **Dog – owner – stick positions?**
- Actions: **Move and move the stick.**
- Agent:
  - Agent state: **Current Position? Do I have the stick?**
  - Policy: **Chase it!**
  - Value function: **???**
  - Model: **???**



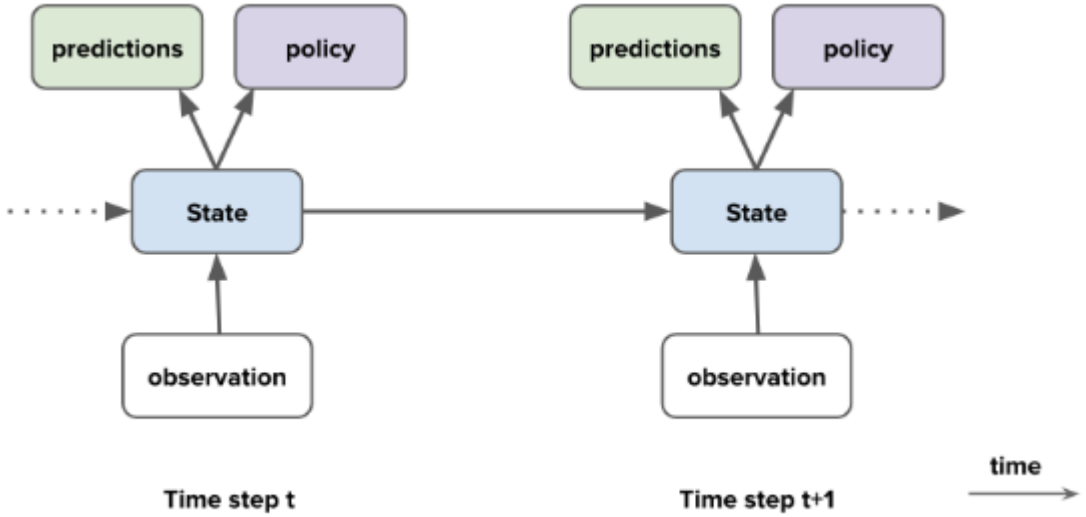
Agent Value Function

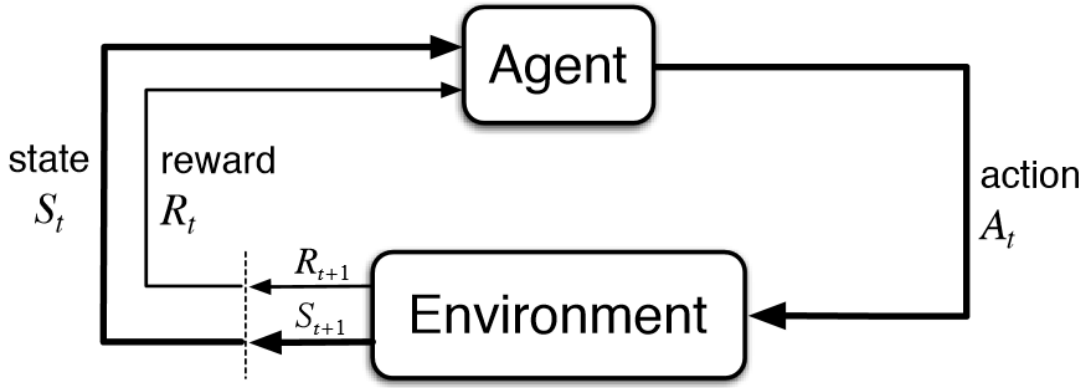
How good is each state and/or action

- Value function is a prediction of future reward.
- Used to evaluate the goodness/badness of states.
- And therefore, to select between actions.

- **Discount factor** trades off importance of immediate vs long-term rewards

- The value depends on a policy.
- Agents often approximate value functions.



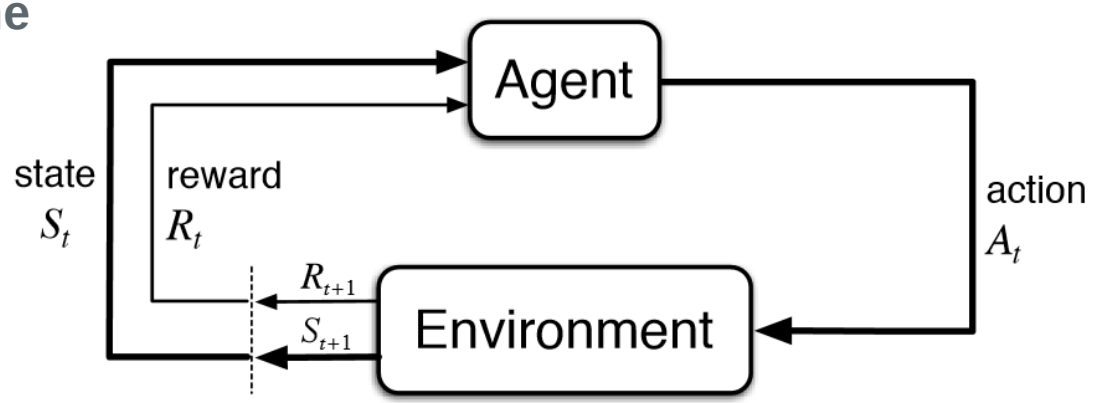


Introductions to Reinforcement Learning

Example: Let's play fetch!

- Reward signal: **Get a treat from owner.**
- Environment: **Dog – owner – stick positions?**
- Actions: **Move the stick.**
- Agent:
  - Agent state: **Current Position? Do I have the stick?**
  - Policy: **Chase it!**
  - Value function: **Do I deserve the treat?**
  - Model: **???**

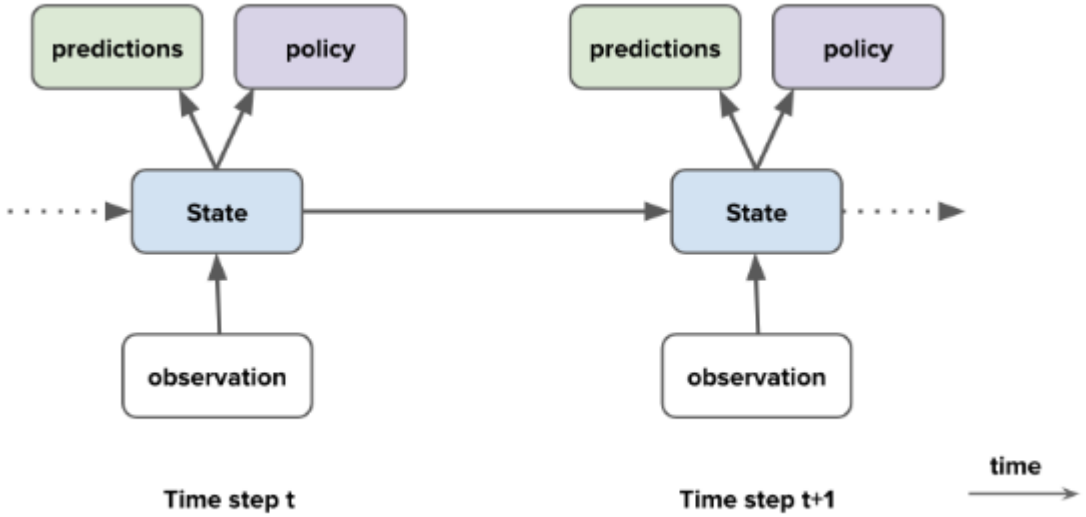
"The bone is a great reward if you can put up with all the knick-knacking and paddy whacking."

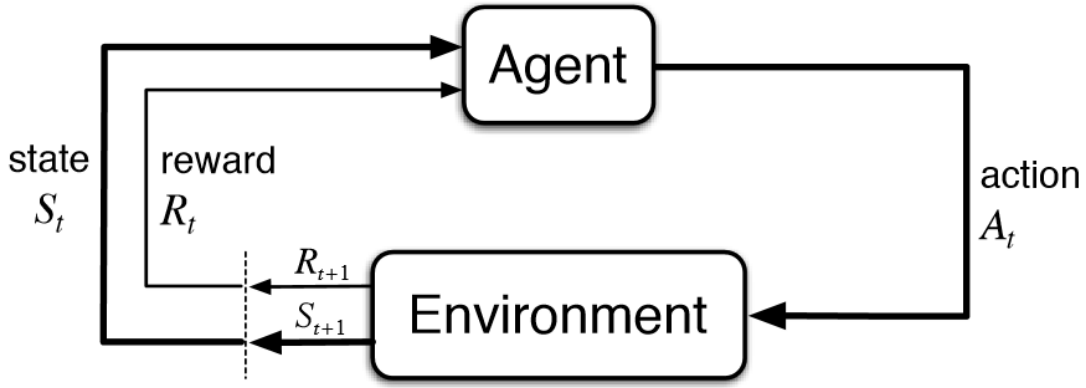


Agent Model

Representation of the environment

- A **model predicts what the environment will do next.**
- Predicts the next state.
- A model does not immediately provide a good policy, there is still a need to plan.

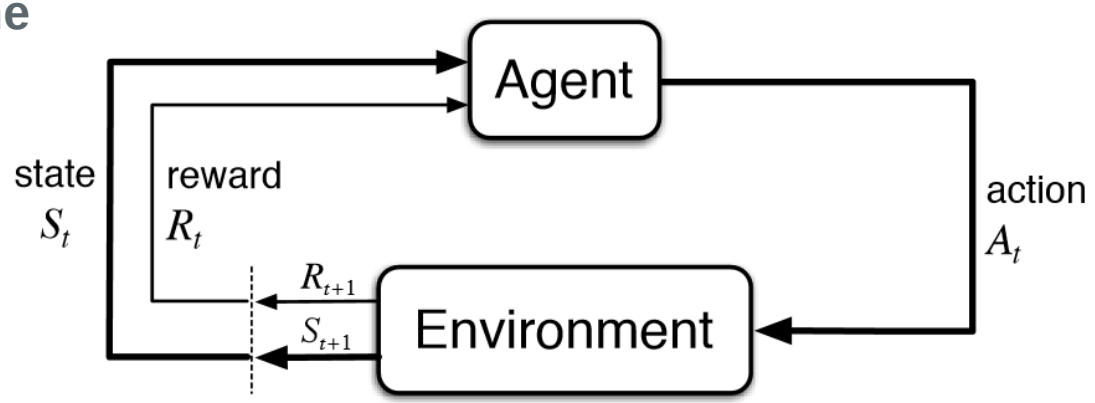




Example: Let's play fetch!

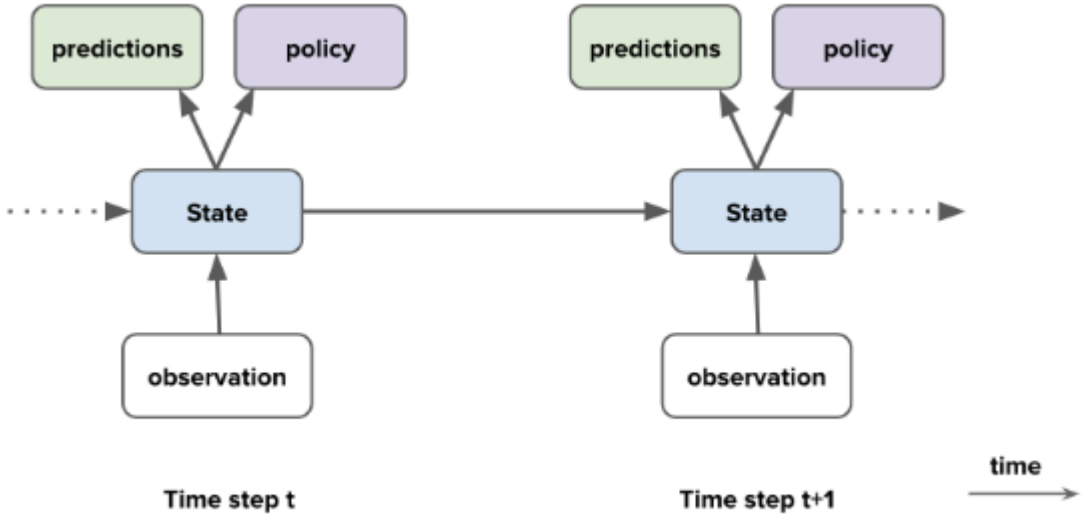
- Reward signal: **Get a treat from owner.**
- Environment: **Dog – owner – stick positions?**
- Actions: **Move and move the stick.**
- Agent:
  - Agent state: **Current Position? Do I have the stick?**
  - Policy: **Chase it!**
  - Value function: **Do I deserve the treat?**
  - Model: **Where will I be next?**

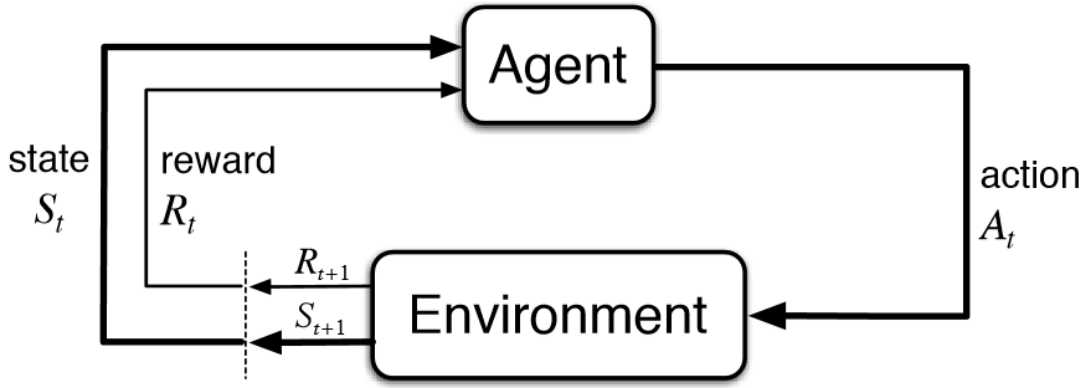
"I started out fetching.



Agent Categories I

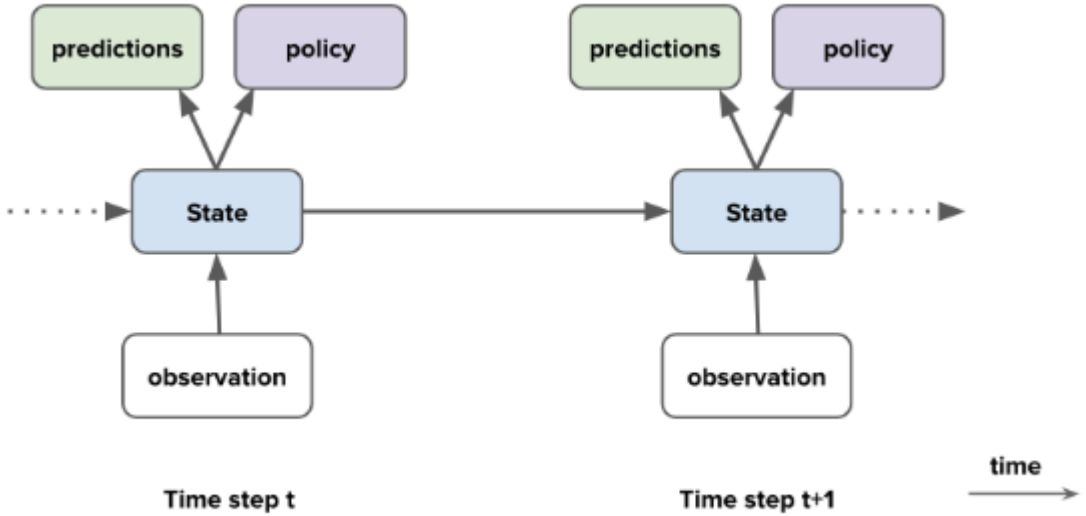
- Value Based
  - No Policy (Implicit)
  - **Value Function**
- Policy Based
  - **Policy**
  - No Value Function
- Actor Critic
  - **Policy**
  - **Value Function**

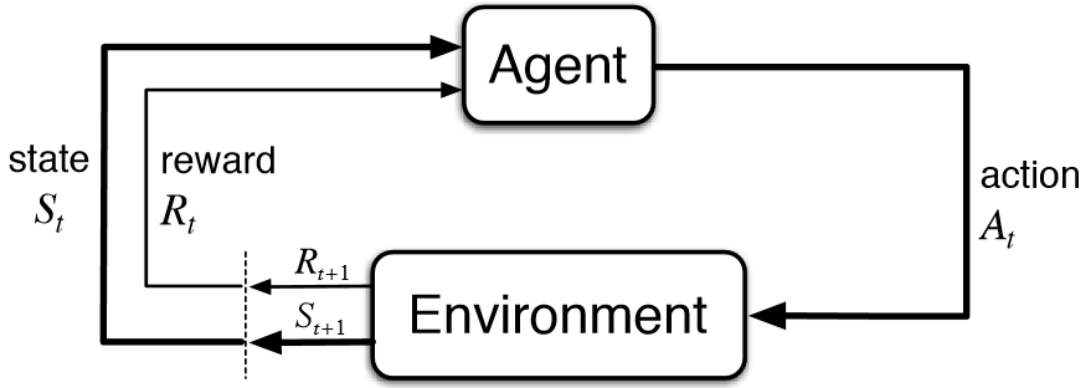




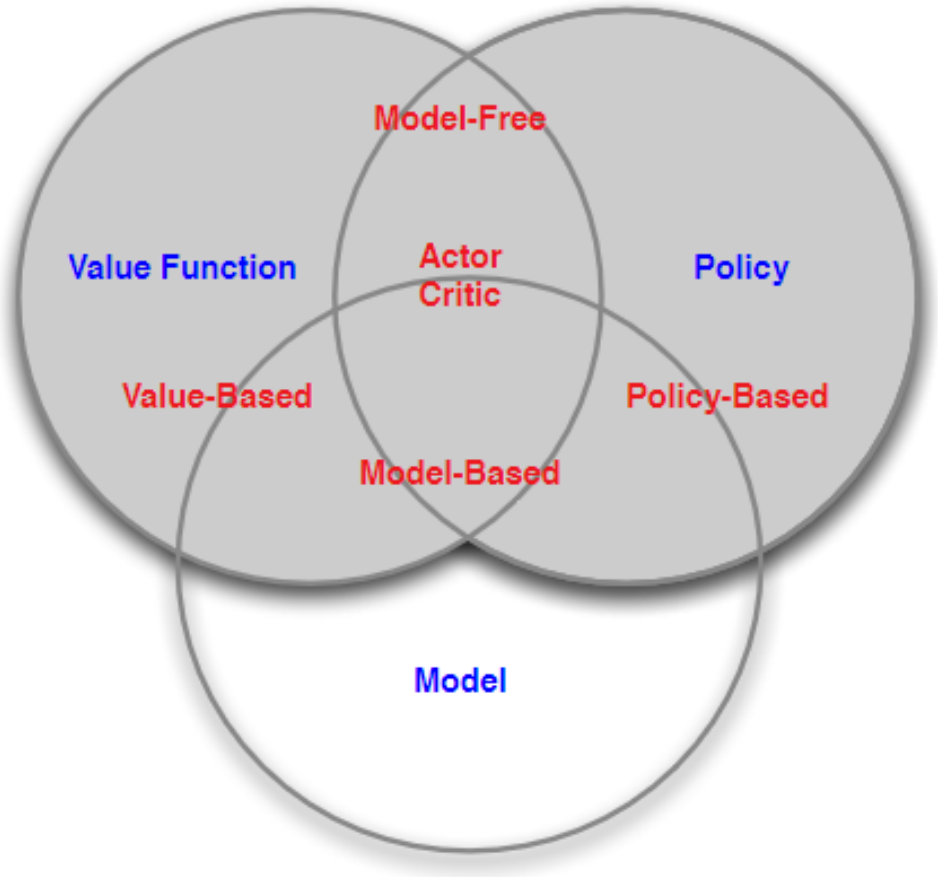
Agent Categories II

- Model Free
  - **Policy and/or Value Function**
  - No Model
- Model Based
  - Optionally Policy and/or Value Function
  - **Model**





Many Concepts of RL



From DeepMind's and UCL with David Silver Course

Introductions to Reinforcement Learning

Prediction/Evaluation vs Control

- **Prediction**: evaluate the future (for a given policy)
- **Control**: optimise the future (find the best policy)
- These can be strongly related.
- If we could predict everything, do we need anything else?

Learning vs Planning

Two fundamental problems in reinforcement learning

- **Learning**:
  - The environment is initially unknown.
  - The agent interacts with the environment.

- **Planning**:
  - A model of the environment is given (or learnt).
  - The agent plans in this model (without external interaction).
  - a.k.a. reasoning, pondering, thought, search, planning.

Exploration vs Exploitation

Two fundamental other problems in reinforcement learning

- **Exploration** finds more information about the environment.
- **Exploitation** exploits known information to maximise reward.
- It is usually important to explore as well as exploit.

- All components are functions
  - **Policies**
  - **Value functions**
  - **Models**
  - **State update**

- Often the **assumptions from supervised learning** (i.i.d., stationarity) are disrupted.

- Deep learning is an important tool.

- **Deep reinforcement learning** is a rich and active research field.

Reinforcement Learning has a lot of Mathematics!

Thank You!

Morada: Campus de Campolide, 1070-312 Lisboa, Portugal
Tel: +351 213 828 610 | Fax: +351 213 828 611

Acreditações e Certificações da NOVA IMS

Cofinanciado por