Introduction

Deep Reinforcement Learning

University of Cambridge

Lecture Outline

1. Announcements
2. Resources
3. Course Content
4. Grading
5. What is RL?

Announcements

- This is my first time teaching
  - Do (privately) give me feedback on the course (sm2558)

Announcements

- This is my first time teaching
  - Do (privately) give me feedback on the course (sm2558)
- This course is structured as interactive lectures
  - Lectures for sessions 1-3, going through RL basics

Announcements

- This is my first time teaching
  - Do (privately) give me feedback on the course (sm2558)
- This course is structured as interactive lectures
  - Lectures for sessions 1-3, going through RL basics
  - If something is unclear, or you have questions or comments stop me and we will discuss them

Announcements

- This is my first time teaching
  - Do (privately) give me feedback on the course (sm2558)
- This course is structured as interactive lectures
  - Lectures for sessions 1-3, going through RL basics
  - If something is unclear, or you have questions or comments stop me and we will discuss them
  - We will go off on tangents if interesting questions arise, please participate!

Announcements

- This is my first time teaching
  - Do (privately) give me feedback on the course (sm2558)
- This course is structured as interactive lectures
  - Lectures for sessions 1-3, going through RL basics
  - If something is unclear, or you have questions or comments stop me and we will discuss them
  - We will go off on tangents if interesting questions arise, please participate!
- This course is very short (3 full lectures)
  - We only have time to focus on one RL algorithm
  - If you are interested in a full course, reach out to the department or try out the Berkeley or UAlberta online courses

Resources

1. Reinforcement Learning, an Introduction (2018, Sutton and Barto)
    - Available for free online (legal)
    - All the RL theory you will ever need

Resources

1. Reinforcement Learning, an Introduction (2018, Sutton and Barto)
    - Available for free online (legal)
    - All the RL theory you will ever need
2. David Silver's slides for his RL course at UCL
    - Builds good intuition
    - Alternative if you don't like my lectures

Resources

1. Reinforcement Learning, an Introduction (2018, Sutton and Barto)
    - Available for free online (legal)
    - All the RL theory you will ever need
2. David Silver's slides for his RL course at UCL
    - Builds good intuition
    - Alternative if you don't like my lectures
3. OpenAI Spinning Up
    - Mixes theory with implementation

Resources

1. Reinforcement Learning, an Introduction (2018, Sutton and Barto)
    - Available for free online (legal)
    - All the RL theory you will ever need
2. David Silver's slides for his RL course at UCL
    - Builds good intuition
    - Alternative if you don't like my lectures
3. OpenAI Spinning Up
    - Mixes theory with implementation
4. CleanRL
    - Verified, single-file implementations of many RL algorithms

Module Goal: Provide a proper understanding of the theoretical foundations of RL

~~**Module Goal**: Provide a proper understanding of the theoretical foundations of RL~~

~~**Module Goal**: Provide a proper understanding of the theoretical foundations of RL~~

Module Goal: Teach you enough to apply RL to solve interesting problems

Course Content

1. Markov Decision Processes
2. Q Learning
3. Student presentations & Miniproject
4. If time permits:
    1. Deep Deterministic Policy Gradient
    2. Partially Observable Processes
    3. Intrinsic Motivation
    4. Multiagent RL
    5. Dreamer/World Models

Grading

1. Attendance (5%)
2. Participation (5%)
3. Presentation and Report (20%)
4. Miniproject (70%)

Presentation and Report

- Presentation and report on an extension of reinforcement learning you find interesting
  - Model-based reinforcement learning
  - Multiagent RL
  - RL applied to chemistry, LLMs, etc
- Presentation and report should:
  - Explain what the topic is and why it's important
  - List a few seminal papers on the topic and briefly summarize each
  - List some promising results and identify where further research is required
  - Propose a research project to further our understanding of the topic
- Due at the start of the final lecture session
- Report ≤ 4 pages
- Presentation length: 10 mins

Miniproject

In-depth project handout given at the end of the final session In a nutshell:

- Implement Deep Q learning in JAX+Equinox that solves a simple video game
  - Free to use external libraries for data collection and storage
  - Must write the training loop and loss functions yourself
- Once working, extend it
  - New tasks (e.g., pixel-based, biology, etc)
  - Improvements (e.g., double q learning, recurrent policies, etc)
- Submit code and a ≤ 4 page write up

Let’s get started

Applications of RL

Reinforcement learning (RL) is a framework for **decision making**.

Applications include:

Applications of RL

Reinforcement learning (RL) is a framework for **decision making**. Applications include:

- Autonomous vehicles

Applications of RL

Reinforcement learning (RL) is a framework for **decision making**.
Applications include:

- Autonomous vehicles
- Video game NPCs

Applications of RL

Reinforcement learning (RL) is a framework for **decision making**.

Applications include:

- Autonomous vehicles
- Video game NPCs
- Alignment in large language models

Applications of RL

Reinforcement learning (RL) is a framework for **decision making**.

Applications include:

- Autonomous vehicles
- Video game NPCs
- Alignment in large language models
- Behavior modeling in psychology/ecology/biology

Applications of RL

Reinforcement learning (RL) is a framework for **decision making**.

Applications include:

- Autonomous vehicles
- Video game NPCs
- Alignment in large language models
- Behavior modeling in psychology/ecology/biology
- Material and drug design

Applications of RL

Reinforcement learning (RL) is a framework for **decision making**.

Applications include:

- Autonomous vehicles
- Video game NPCs
- Alignment in large language models
- Behavior modeling in psychology/ecology/biology
- Material and drug design
- Finance

Applications of RL

Reinforcement learning (RL) is a framework for **decision making**.

Applications include:

- Autonomous vehicles
- Video game NPCs
- Alignment in large language models
- Behavior modeling in psychology/ecology/biology
- Material and drug design
- Finance
- Artificial General Intelligence?

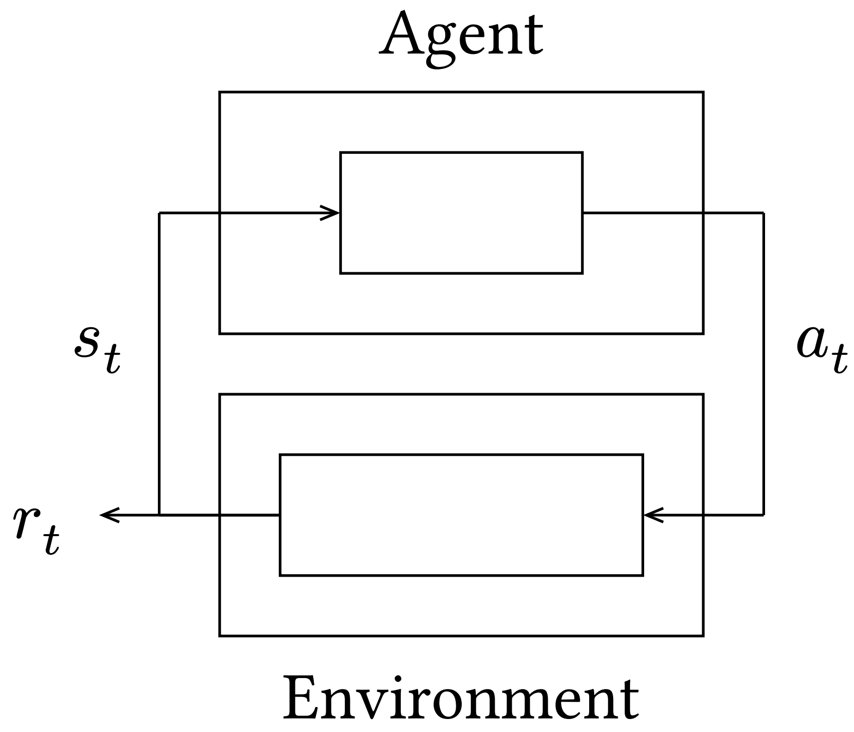
Applications of RL

Reinforcement learning (RL) is a framework for **decision making**.

Applications include:

- Autonomous vehicles
- Video game NPCs
- Alignment in large language models
- Behavior modeling in psychology/ecology/biology
- Material and drug design
- Finance
- Artificial General Intelligence?
- Anywhere with cause and effect
  - Where you **change** the world by **interacting** with it

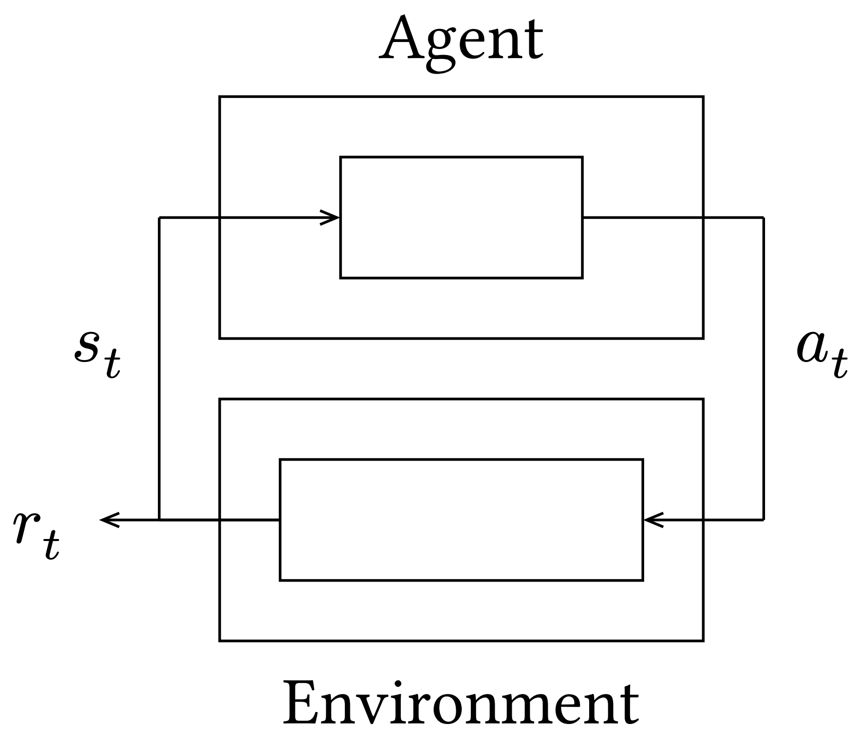
The Agent and Environment



s_t: state, a_t: action, r_t: reward

- In RL, we have the **agent** and **environment**

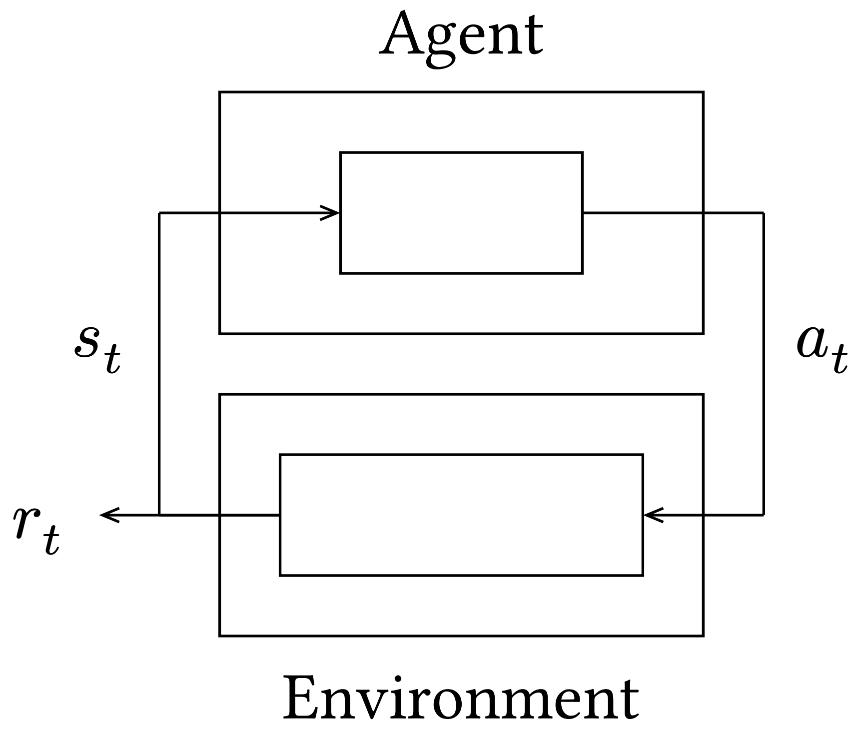
The Agent and Environment



s_t: state, a_t: action, r_t: reward

- In RL, we have the **agent** and **environment**
- The agent takes **actions** in the environment

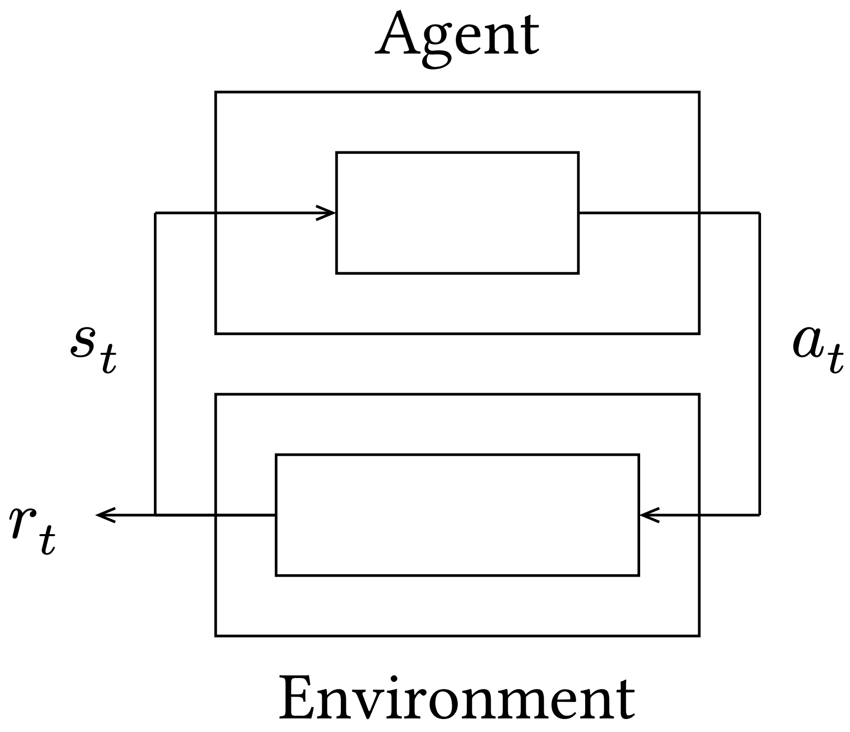
The Agent and Environment



s_t: state, a_t: action, r_t: reward

- In RL, we have the **agent** and **environment**
- The agent takes **actions** in the environment
- Actions change the environment **state**, producing an new state and **reward**

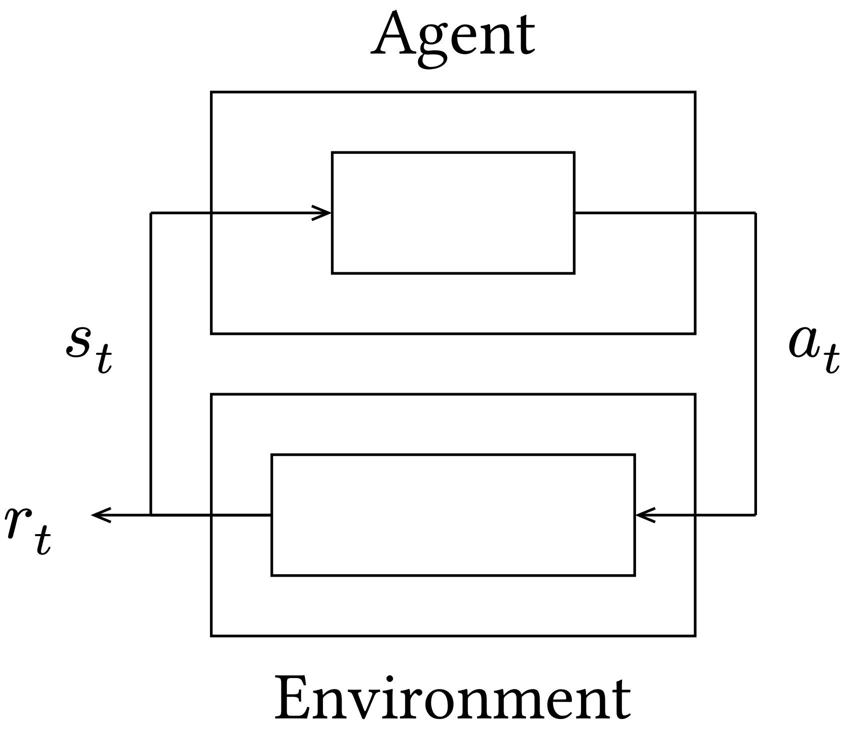
The Agent and Environment



s_t: state, a_t: action, r_t: reward

- In RL, we have the **agent** and **environment**
- The agent takes **actions** in the environment
- Actions change the environment **state**, producing an new state and **reward**
- The cycle continues for $t = 0, 1, \ldots$

The Agent and Environment



s_t: state, a_t: action, r_t: reward

- In RL, we have the **agent** and **environment**
- The agent takes **actions** in the environment
- Actions change the environment **state**, producing an new state and **reward**
- The cycle continues for $t = 0, 1, \ldots$
- Goal is to maximize the **cumulative reward**

Questions?

Exercise

Can anybody come up with a real world problem that lends itself to RL?

- Agent taking actions in an environment, in search of some reward
- What is the agent, and what is the environment? The reward?

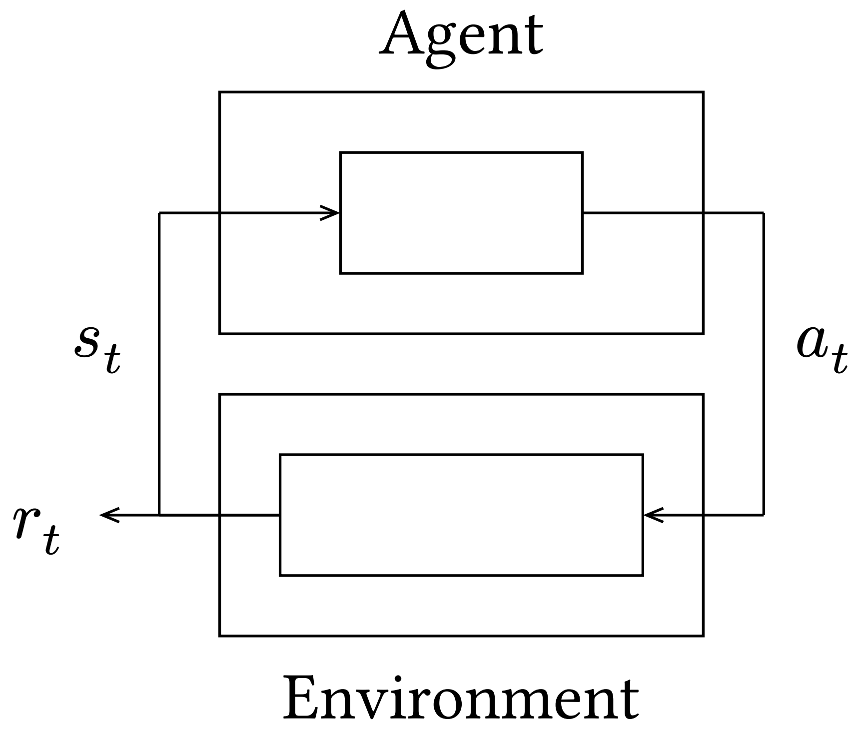
UNIVERSITY OF CAMBRIDGE

Markov Decision Processes

Deep Reinforcement Learning

University of Cambridge

Last Time



s_t: state, a_t: action, r_t: reward

Markov Decision Processes

By definition, RL solves **Markov Decision Processes (MDPs)**.

Markov Decision Processes

By definition, RL solves **Markov Decision Processes (MDPs)**.

To solve a problem using RL, we first must convert the problem into an MDP. We call this converted problem the **environment**.

Markov Decision Processes

By definition, RL solves **Markov Decision Processes (MDPs)**.

To solve a problem using RL, we first must convert the problem into an MDP. We call this converted problem the **environment**.

How you structure your problem is **critical** – more important than which algorithms you use, how much compute you have, etc.

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

$T : S \times A \rightarrow \Delta S$ The **state transition function**.

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

$T : S \times A \rightarrow \Delta S$ The **state transition function**.

$R : S \rightarrow \mathbb{R}$ is the **reward function**.

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

$T : S \times A \to \Delta S$ The **state transition function**.

$R : S \to \mathbb{R}$ is the **reward function**.

$\gamma \in [0, 1]$ is a discount factor that we will explain later

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

$T : S \times A \rightarrow \Delta S$ The **state transition function**.

$R : S \rightarrow \mathbb{R}$ is the **reward function**.

$\gamma \in [0, 1]$ is a discount factor that we will explain later

Let us briefly explain these terms.

State Space

S is the set of states known as the **state space**.

State Space

S is the set of states known as the **state space**.

Recall that the environment is always changing

State Space

S is the set of states known as the **state space**.

Recall that the environment is always changing

We need a way to describe what state the environment is in

State Space

S is the set of states known as the **state space**.

Recall that the environment is always changing

We need a way to describe what state the environment is in

If the environment is a table, the state space might describe the positions of all objects on the table $\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \dots \end{bmatrix}$

State Space

S is the set of states known as the **state space**.

Recall that the environment is always changing

We need a way to describe what state the environment is in

If the environment is a table, the state space might describe the positions of all objects on the table $\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \dots \end{bmatrix}$

Action Space

A is the set of actions known as the **action space**

Action Space

A is the set of actions known as the **action space**

What capabilities does the agent have?

Action Space

A is the set of actions known as the **action space**

What capabilities does the agent have?

For the table example, I can apply a force to a specific object on the table

$$[F_x \;\; F_y \;\; i]$$

Action Space

A is the set of actions known as the **action space**

What capabilities does the agent have?

For the table example, I can apply a force to a specific object on the table

$$\begin{bmatrix} F_x & F_y & i \end{bmatrix}$$

State Transition Function

$T : S \times A \to \Delta S$ The **state transition function**.

State Transition Function

$T : S \times A \to \Delta S$ The **state transition function**.

Sometimes called “transition dynamics”, “state transitions matrix”, etc

State Transition Function

$T : S \times A \to \Delta S$ The **state transition function**.

Sometimes called "transition dynamics", "state transitions matrix", etc

"Rules" of the environment, determine the (stochastic) evolution of the environment

State Transition Function

$T : S \times A \rightarrow \Delta S$ The **state transition function**.

Sometimes called "transition dynamics", "state transitions matrix", etc

"Rules" of the environment, determine the (stochastic) evolution of the environment

$$T\left(\underbrace{\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \ldots \end{bmatrix}}_{\text{state}}, \underbrace{\begin{bmatrix} F_x & F_y & i \end{bmatrix}}_{\text{action}}\right) = \underbrace{\Delta\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \ldots \end{bmatrix}}_{\text{new state}}$$

State Transition Function

$T : S \times A \rightarrow \Delta S$ The **state transition function**.

Sometimes called "transition dynamics", "state transitions matrix", etc

"Rules" of the environment, determine the (stochastic) evolution of the environment

$$T\left(\underbrace{\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \dots \end{bmatrix}}_{\text{state}}, \underbrace{\begin{bmatrix} F_x & F_y & i \end{bmatrix}}_{\text{action}}\right) = \underbrace{\Delta\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \dots \end{bmatrix}}_{\text{new state}}$$

This is a **Markov** decision process because transition dynamics are **conditionally independent** of past states and actions

$$T(s_t, a_t \mid s_{t-1}, a_{t-1}, \dots, s_0, a_0) = T(s_t, a_t)$$

Reward Function

$R: S \to \mathbb{R}$ is the **reward function**.

Reward Function

$R : S \to \mathbb{R}$ is the **reward function**.

Produces reward based on the state

Reward Function

$R : S \to \mathbb{R}$ is the **reward function**.

Produces reward based on the state

Reward function determines agent behavior

Reward Function

$R : S \to \mathbb{R}$ is the **reward function**.

Produces reward based on the state

Reward function determines agent behavior

+100 for pushing objects onto the floor, or +100 for pushing objects to the centre

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

$T : S \times A \to \Delta S$ The **state transition function**.

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

$T : S \times A \to \Delta S$ The **state transition function**.

$R : S \to \mathbb{R}$ is the **reward function**.

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

$T : S \times A \to \Delta S$ The **state transition function**.

$R : S \to \mathbb{R}$ is the **reward function**.

$\gamma \in [0, 1]$ is a discount factor that we will explain later

Markov Decision Processes

Definition: An MDP is a 5-tuple (S, A, T, R, γ)

S is the set of states known as the **state space**.

A is the set of actions known as the **action space**

$T : S \times A \rightarrow \Delta S$ The **state transition function**.

$R : S \rightarrow \mathbb{R}$ is the **reward function**.

$\gamma \in [0, 1]$ is a discount factor that we will explain later

State Transition Function

$T : S \times A \rightarrow \Delta S$ The **state transition function**.

State Transition Function

$T : S \times A \to \Delta S$ The **state transition function**.

$$T\left(\underbrace{\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \dots \end{bmatrix}}_{\text{state}}, \underbrace{\begin{bmatrix} F_x & F_y & i \end{bmatrix}}_{\text{action}}\right) = \underbrace{\Delta\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \dots \end{bmatrix}}_{\text{new state}}$$

State Transition Function

$T : S \times A \to \Delta S$ The **state transition function**.

$$T\left(\underbrace{\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \dots \end{bmatrix}}_{\text{state}}, \underbrace{\begin{bmatrix} F_x & F_y & i \end{bmatrix}}_{\text{action}}\right) = \underbrace{\Delta\begin{bmatrix} x_1 & y_1 & x_2 & y_2 & \dots \end{bmatrix}}_{\text{new state}}$$

This is a **Markov** decision process because transition dynamics are **conditionally independent** of past states and actions

$$T(s_t, a_t \mid s_{t-1}, a_{t-1}, \dots, s_0, a_0) = T(s_t, a_t)$$

Key Concept: To be **Markov**, state must contain sufficient information to predict next state

Super Mario Bros

- **State Space (S)?**

Super Mario Bros

- **State Space (S)**?
  - The position and velocity $(x, y, \dot{x}, \dot{y})$ of Mario and Goombas
  - The score
  - Number of coins collected
  - The time remaining
  - Which question blocks have been opened
  - Which goombas have been squished

Super Mario Bros

- **State Space (S)?**
  - The position and velocity $(x, y, \dot{x}, \dot{y})$ of Mario and Goombas
  - The score
  - Number of coins collected
  - The time remaining
  - Which question blocks have been opened
  - Which goombas have been squished

$$S = \left\{\mathbb{R}^4, \mathbb{R}^4, ..., \mathbb{Z}_+, \mathbb{Z}_+, \mathbb{Z}_+, \{0, 1\}^k\right\}$$

Super Mario Bros

- **State Space (S)?**

Super Mario Bros

- **State Space (S)?**

2x256x240x3 pixels.

$$\text{E.g.} \begin{pmatrix} \begin{pmatrix} 255 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 170 \\ 10 \\ 50 \end{pmatrix} & \cdots \\ \begin{pmatrix} 10 \\ 100 \\ 235 \end{pmatrix} & \begin{pmatrix} 200 \\ 200 \\ 35 \end{pmatrix} & \cdots \\ \vdots & & \ddots \end{pmatrix}, \begin{pmatrix} \begin{pmatrix} 255 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 170 \\ 10 \\ 50 \end{pmatrix} & \cdots \\ \begin{pmatrix} 10 \\ 100 \\ 235 \end{pmatrix} & \begin{pmatrix} 200 \\ 200 \\ 35 \end{pmatrix} & \cdots \\ \vdots & & \ddots \end{pmatrix}$$

Super Mario Bros

- **State Space (S)?**

2x256x240x3 pixels.

E.g. $$\begin{pmatrix} \begin{pmatrix} 255 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 170 \\ 10 \\ 50 \end{pmatrix} & \cdots \\ \begin{pmatrix} 10 \\ 100 \\ 235 \end{pmatrix} & \begin{pmatrix} 200 \\ 200 \\ 35 \end{pmatrix} & \cdots \\ \vdots & & \ddots \end{pmatrix}, \begin{pmatrix} \begin{pmatrix} 255 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 170 \\ 10 \\ 50 \end{pmatrix} & \cdots \\ \begin{pmatrix} 10 \\ 100 \\ 235 \end{pmatrix} & \begin{pmatrix} 200 \\ 200 \\ 35 \end{pmatrix} & \cdots \\ \vdots & & \ddots \end{pmatrix}$$

Two images necessary to compute velocities!

Super Mario Bros

- **State Space (S)?**

2x256x240x3 pixels.

$$\text{E.g.} \begin{pmatrix} \begin{pmatrix} 255 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 170 \\ 10 \\ 50 \end{pmatrix} & \cdots \\ \begin{pmatrix} 10 \\ 100 \\ 235 \end{pmatrix} & \begin{pmatrix} 200 \\ 200 \\ 35 \end{pmatrix} & \cdots \\ \vdots & & \ddots \end{pmatrix}, \begin{pmatrix} \begin{pmatrix} 255 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 170 \\ 10 \\ 50 \end{pmatrix} & \cdots \\ \begin{pmatrix} 10 \\ 100 \\ 235 \end{pmatrix} & \begin{pmatrix} 200 \\ 200 \\ 35 \end{pmatrix} & \cdots \\ \vdots & & \ddots \end{pmatrix}$$

Two images necessary to compute velocities!

$$S = \mathbb{Z}_{<255}^{2 \times 256 \times 240 \times 3}$$

Super Mario Bros

- **Action Space (A)?**

Super Mario Bros

- **Action Space (A)**?
  - Acceleration of Mario $\ddot{x}$

Super Mario Bros

- **Action Space (A)?**
  - Acceleration of Mario $\ddot{x}$
    - But when playing Mario, we cannot explicitly set $\ddot{x}$

Super Mario Bros

- **Action Space (A)?**

Super Mario Bros

- **Action Space (A)**?
  - The Nintendo controller has A, B, up, down, left, right buttons
    - $A = \{A, B, \text{up}, \text{down}, \text{left}, \text{right}\}$

Super Mario Bros

- **Action Space (A)**?
  - The Nintendo controller has A, B, up, down, left, right buttons
    - $A = \{A, B, \text{up}, \text{down}, \text{left}, \text{right}\}$
      - Cannot represent pressing multiple buttons at once

Super Mario Bros

- **Action Space (A)?**
  - The Nintendo controller has A, B, up, down, left, right buttons
    - $A = \{A, B, \text{up}, \text{down}, \text{left}, \text{right}\}$
      - Cannot represent pressing multiple buttons at once
    - $A = \{0, 1\}^5$