

Reinforcement Learning

Some slides were adapted/taken from various sources, including Prof. Andrew Ng's Coursera Lectures, Stanford University, Prof. Kilian Q. Weinberger's lectures on Machine Learning, Cornell University, Prof. Sudeshna Sarkar's Lecture on Machine Learning, IIT Kharagpur, Prof. Bing Liu's lecture, University of Illinois at Chicago (UIC), CS231n: Convolutional Neural Networks for Visual Recognition lectures, Stanford University, Dr. Luis Serrano, Prof. Alexander Ihler and many more. We thankfully acknowledge them. Students are requested to use this material for their study only and **NOT** to distribute it.

Outlines

- What is Reinforcement Learning?
- Markov Decision Processes
- Q-Learning
- Policy Gradients

Supervised Learning

Data: (x, y)

x is data, y is label

Goal: Learn a *function* to map

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.



→ Cat

Classification

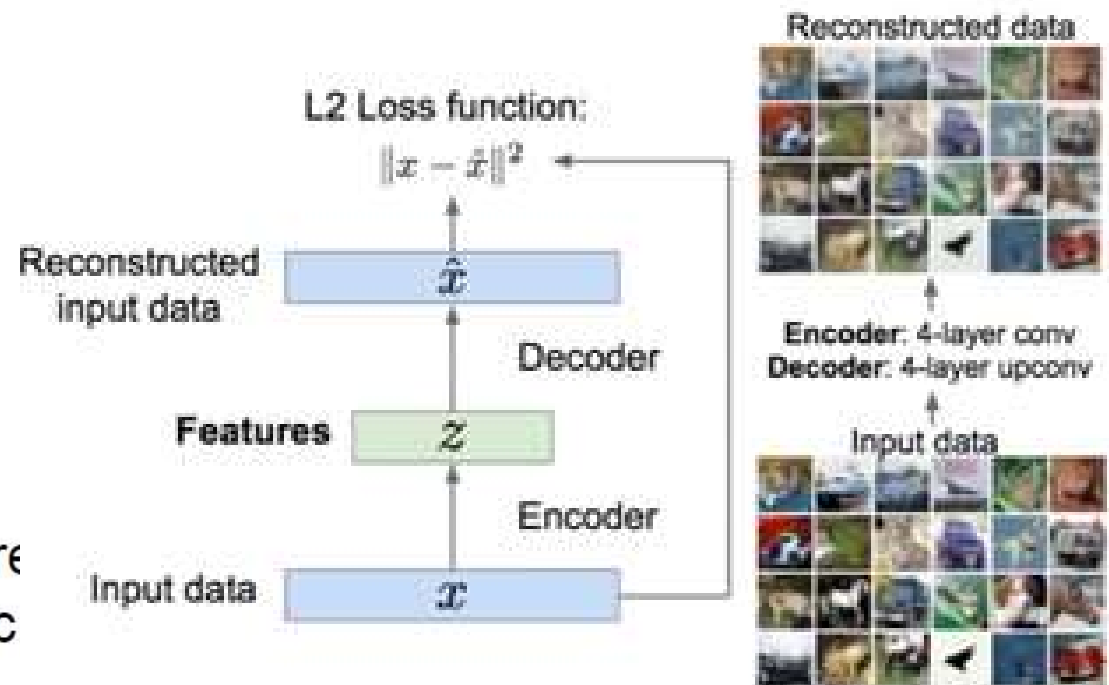
Un-Supervised Learning

Data: x

Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

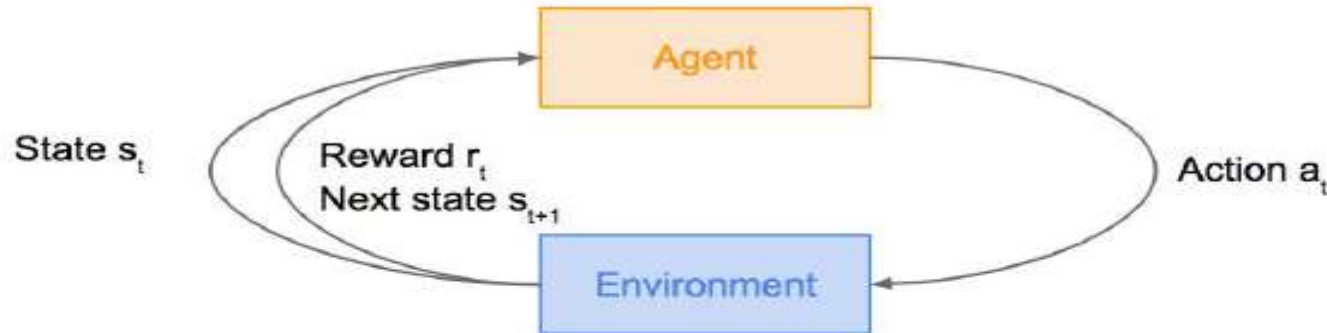
Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc



Reinforcement Learning

Problems involving an **agent** interacting with an **environment**, which provides numeric **reward** signals

Goal: Learn how to take actions in order to maximize reward



Reinforcement Learning

Agent

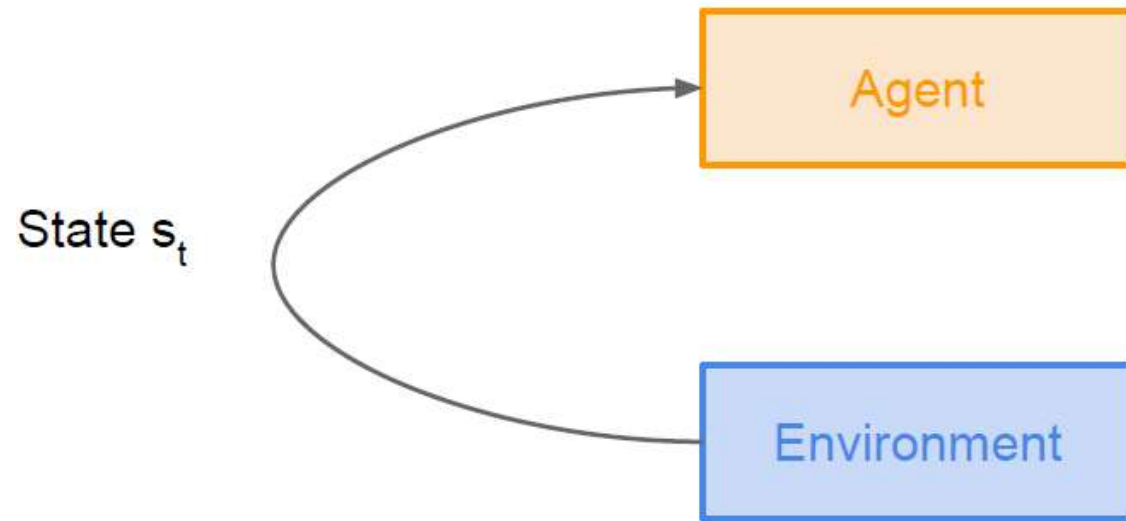


```
graph TD; Agent[Agent] --> Environment[Environment]; Environment --> Agent;
```

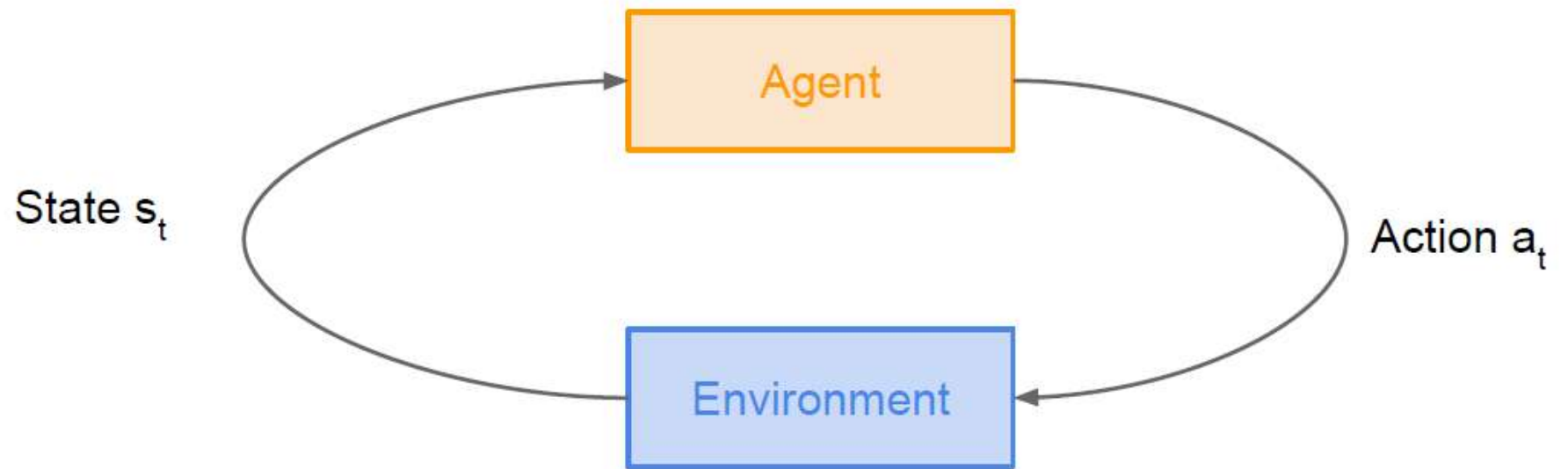
The diagram illustrates the Reinforcement Learning loop. It consists of two main components: the Agent and the Environment. The Agent is represented by an orange box at the top, and the Environment is represented by a blue box at the bottom. Arrows indicate the flow of information: the Agent sends actions to the Environment, and the Environment sends observations and rewards back to the Agent.

Environment

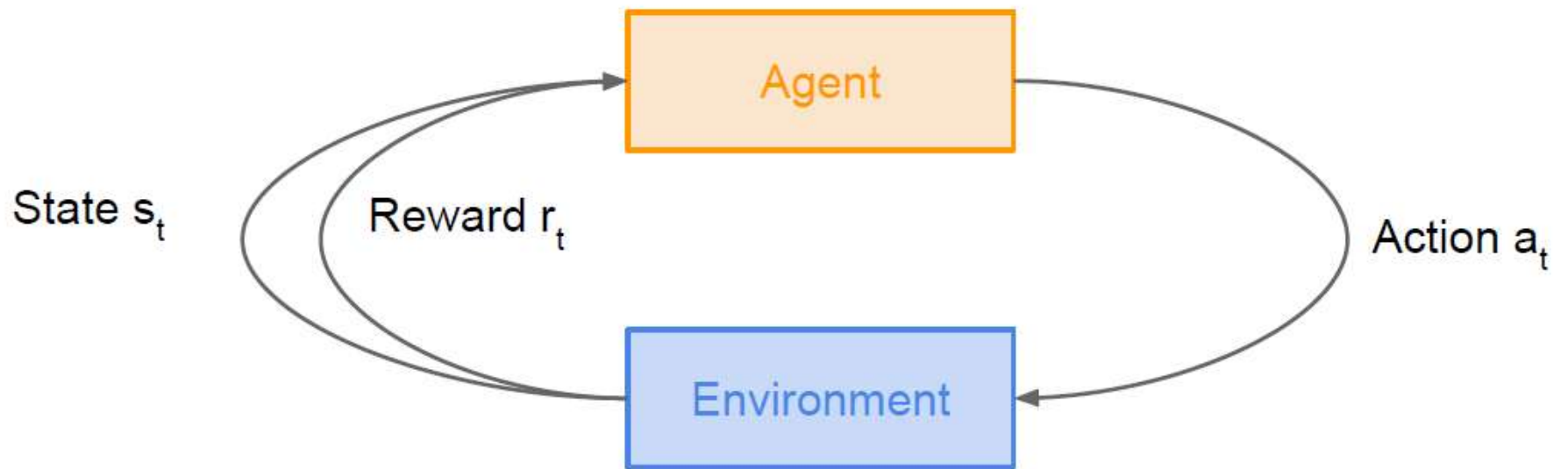
Reinforcement Learning



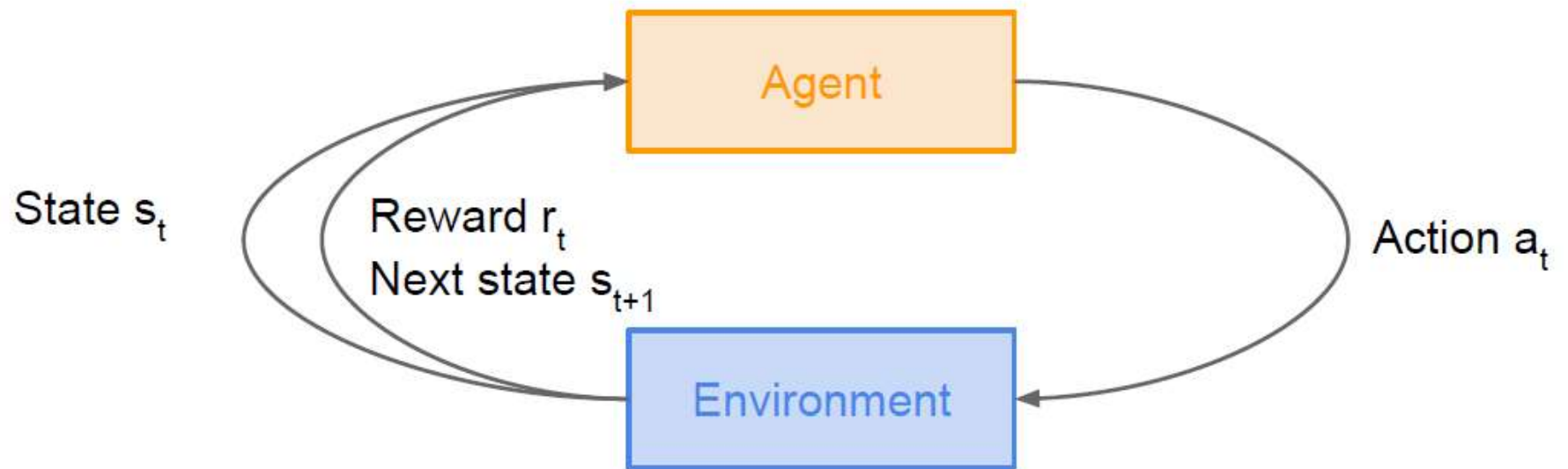
Reinforcement Learning



Reinforcement Learning



Reinforcement Learning



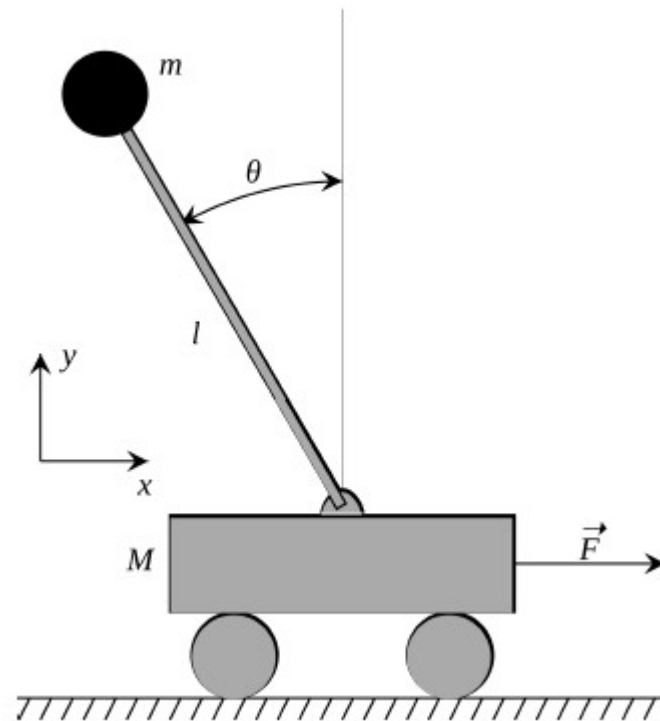
Cart-Pole Problem

Objective : Balance a pole on top of a movable cart

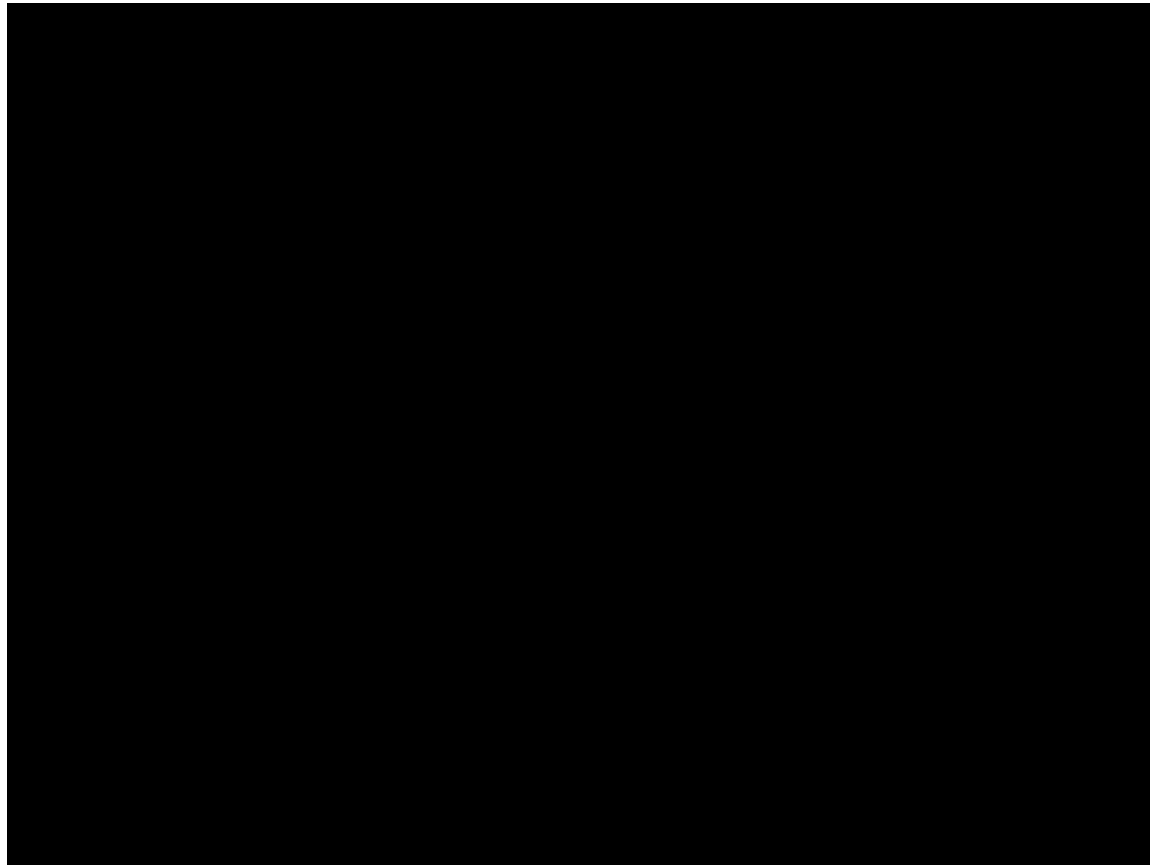
State : angle, angular speed, position, horizontal velocity

Action : horizontal force applied on the cart

Reward : 1 at each time step if the pole is upright



Robot Locomotion

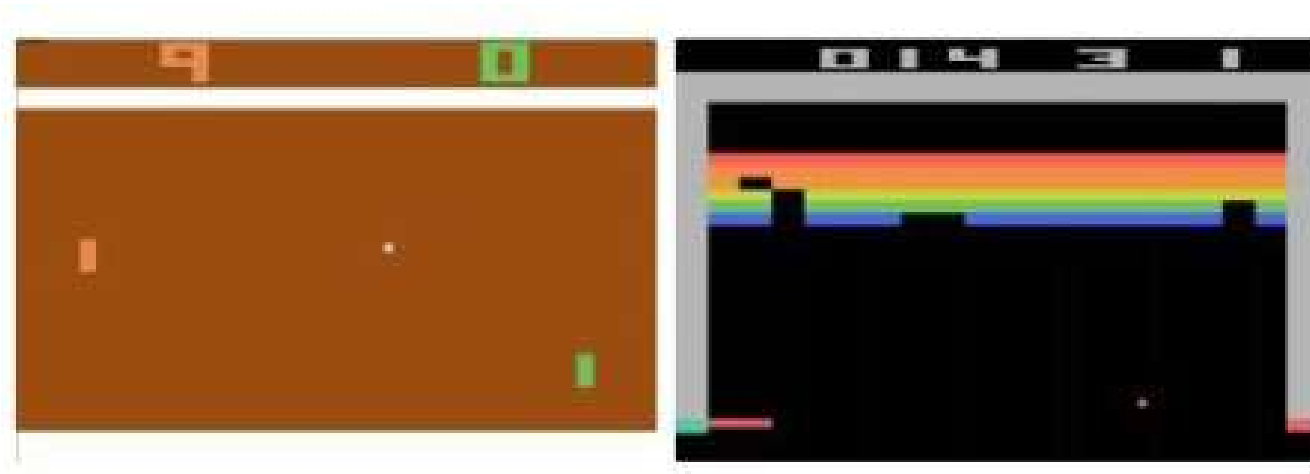


Objective: Make the robot move forward State: Angle and position of the joints

Action: Torques applied on joints

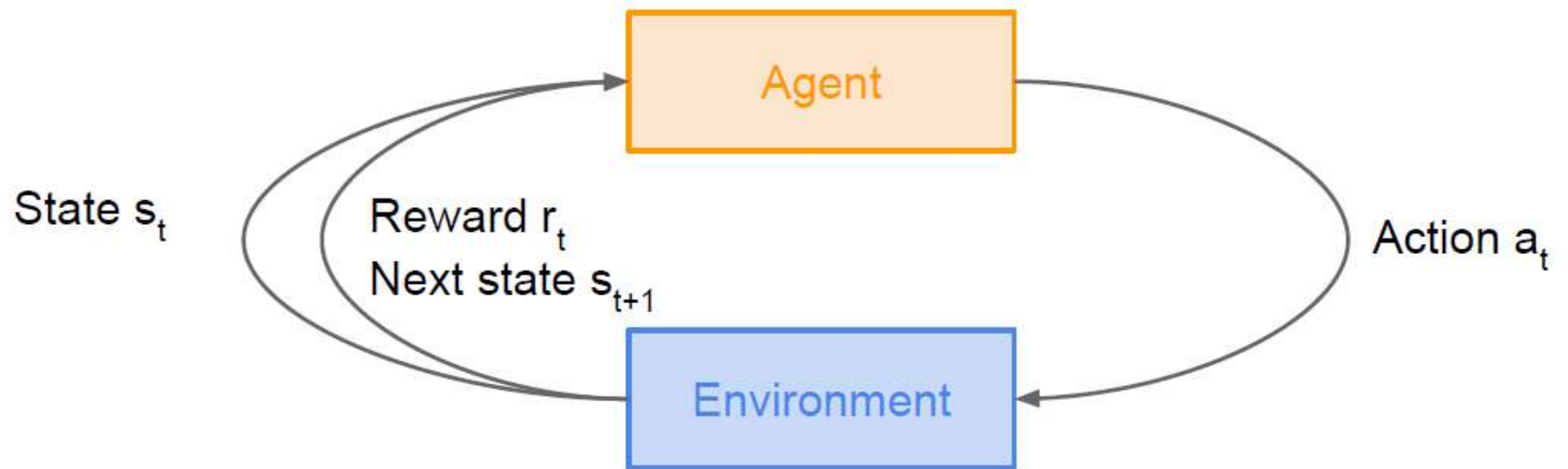
Reward: 1 at each time step upright + forward movement

Atari Games



- Objective** : Complete the game with the highest score
- State** : Raw pixel inputs of the game state
- Action** : Game controls e.g. Left, Right, Up, Down
- Reward** : Score increase/decrease at each time step

RL: Mathematical Formulation



Markov Decision Process

- Mathematical formulation of the RL problem
- **Markov property:** Current state completely characterises the state of the world

Defined by: $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{P}, \gamma)$

\mathcal{S} : set of possible states

\mathcal{A} : set of possible actions

\mathcal{R} : distribution of reward given (state, action) pair

\mathbb{P} : transition probability i.e. distribution over next state given (state, action) pair

γ : discount factor

Markov Decision Process

S is a set of a finite state that describes the environment.

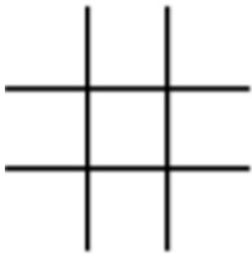
A is a set of a finite actions that describes the action that can be taken by the agent

P is a probability matrix that tells the probability of moving from one state to the other.

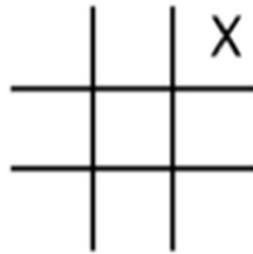
R is a set of rewards that depend on the state and the action taken. Rewards are not necessarily positive, they should be seen as outcome of an action done by the agent when it is at a certain state. So negative reward indicates bad result, whereas positive reward indicates good result.

γ is a discount factor, that tells how important future rewards are to the current state. Discount factor is a value between 0 and 1. A reward R that occurs N steps in the future from the current state, is multiplied by γ^N to describe its importance to the current state. For example consider $\gamma = 0.9$ and a reward $R = 10$ that is 3 steps ahead of our current state. The importance of this reward to us from where we stand is equal to $(0.9^3) * 10 = 7.29$.

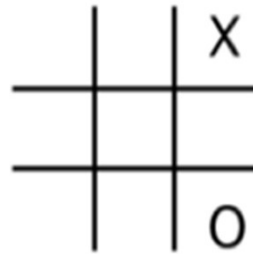
Example: Tic-tac-toe



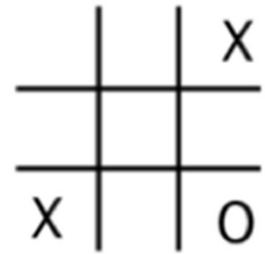
Before
game
begins



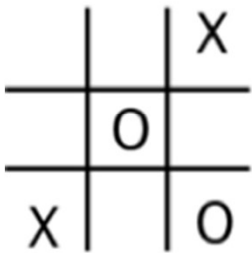
X's
first
move



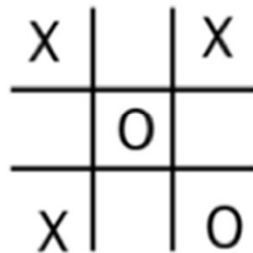
O's
first
move



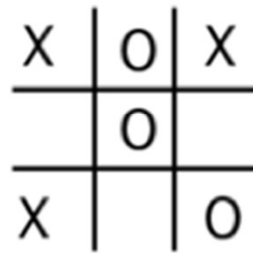
X's
second
move



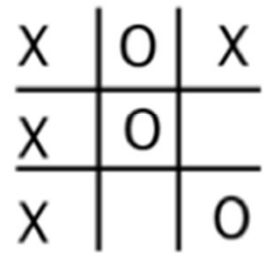
O's
second
move



X's
third
move



O's
third
move



X wins on
X's fourth
move

Man Vs Machine

Deep Blue
IBM Super Computer



Garry Kasparov
World Chess Champion

First match February 10, 1996: took place in [Philadelphia, Pennsylvania](#)

Result: **Kasparov**–Deep Blue (4–2)

Record set: First computer program to defeat a world champion in a *[classical](#) game* under tournament regulations

Second match (rematch)

May 11, 1997: held in [New York City, New York](#)

Result: **Deep Blue**–Kasparov (3½–2½)

Record set: First computer program to defeat a world champion in a *match* under tournament regulations

Thanks