ELTE | FACULTY OF INFORMATICS

# REINFORCEMENT LEARNING INTRODUCTION

Deep Reinforcement Learning
Balázs Nagy, PhD

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

# References


- Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.
- © Alexander Amini and Ava Amini
  MIT 6.S191: Introduction to Deep Learning
  http://introtodeeplearning.com/


**These slides are in large part based on the Sutton & Barto book**

# What is Reinforcement Learning (RL)?

# What is Reinforcement Learning (RL)?

- Examples:
  - Infant plays (no explicit teacher)
  - Teach a dog new trick (with teacher)

# What is Reinforcement Learning (RL)?

- Examples:
  - Infant plays (no explicit teacher)
  - Teach a dog new trick (with teacher)

**Reinforcement Learning is the closest what humans and animals do**

# What is Reinforcement Learning (RL)?

- Computational approach to learning from interactions
  - Explore idealized learning situations
  - Evaluate the effectiveness of various learning methods
  - Goal directed learning from interactions

# What is Reinforcement Learning (RL)?

- Computational approach to learning from interactions
  - Explore idealized learning situations
  - Evaluate the effectiveness of various learning methods
  - Goal directed learning from interactions

**RL: learning what to do – how to map situations to actions – so as to maximize a numerical reward signal**

- Features:
  - Trial-and-error search
  - Delayed reward

# What is Reinforcement Learning (RL)?

- Reinforcement Learning
  - Problem
  - Class of solutions (works well on the problem)
  - Field (studies the problem and its solution)

# What is Reinforcement Learning (RL)?

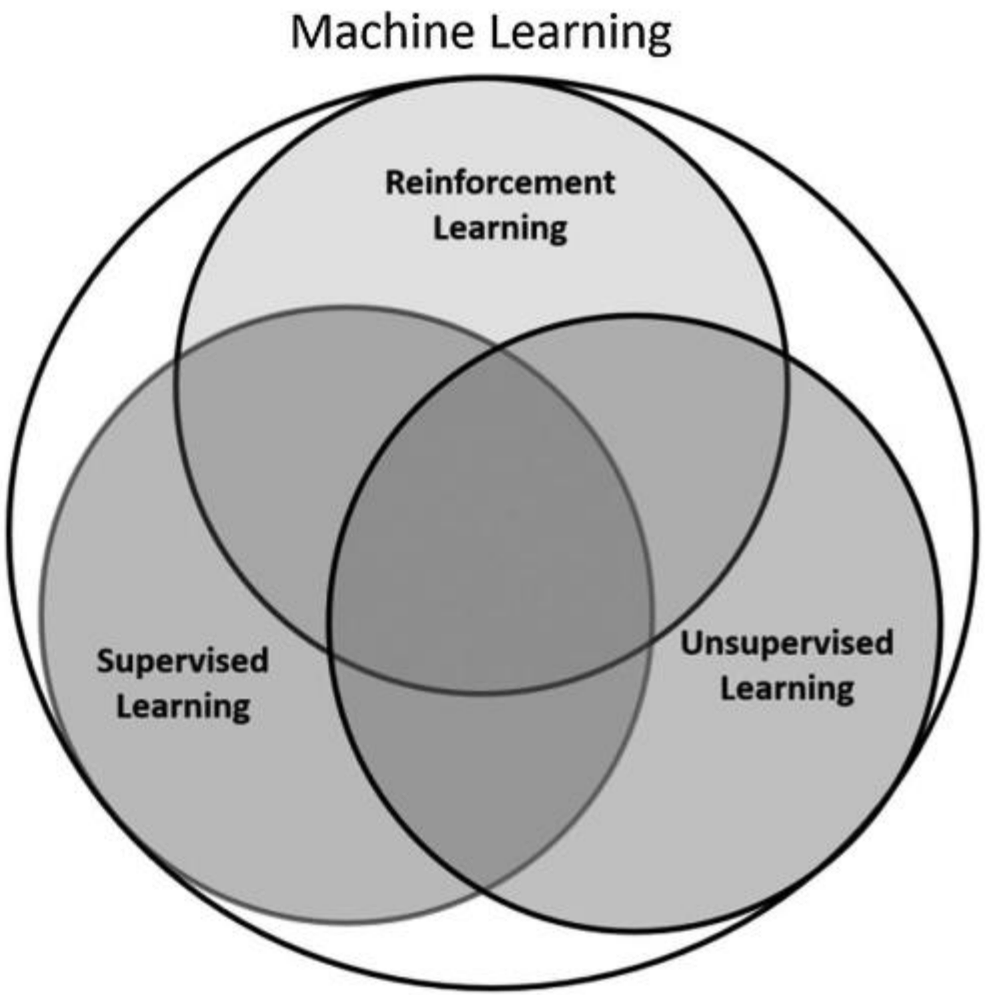
- Reinforcement Learning
  - Problem
  - Class of solutions (works well on the problem)
  - Field (studies the problem and its solution)

**Same name for 3 conceptually separate things**

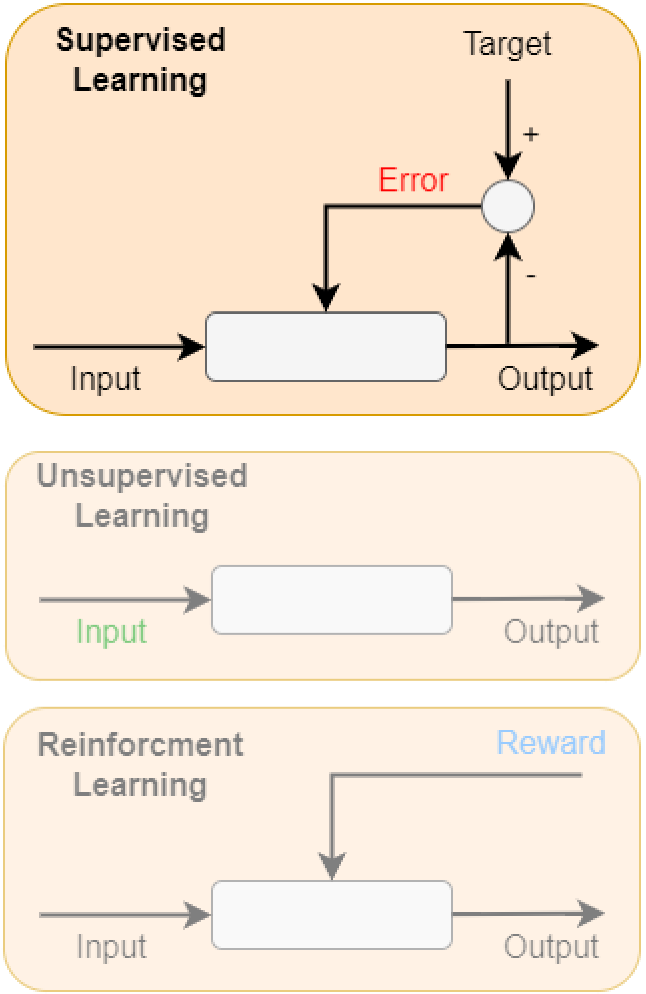
**Artificial Intelligence**
*Any technique that enables computers to mimic human intelligence.*

**Machine Learning**
*A subset of AI that includes abstruse statistical techniques that enable machines to improve at tasks with experience.*

**Deep Learning**
*The subset of machine learning composed of algorithms that permit software to train itself to perform tasks by exposing multilayered neural networks to vast amount of data.*

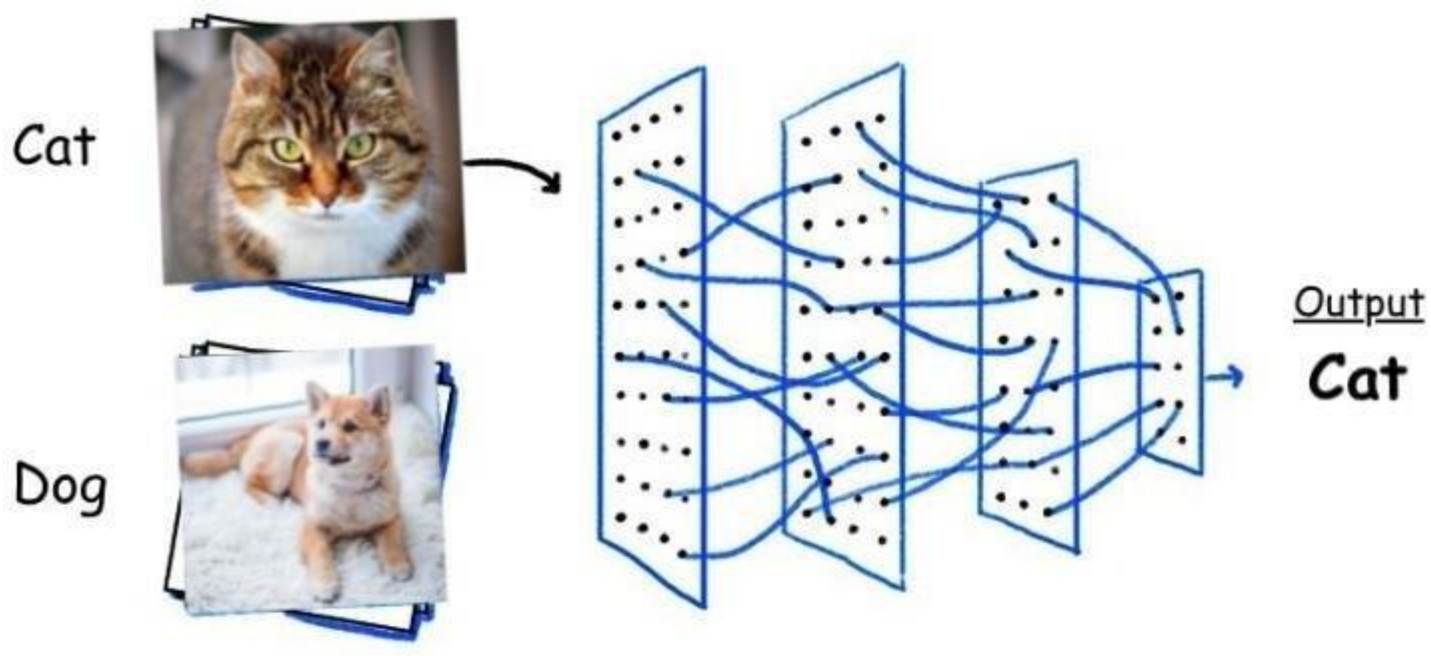


# Learning methods

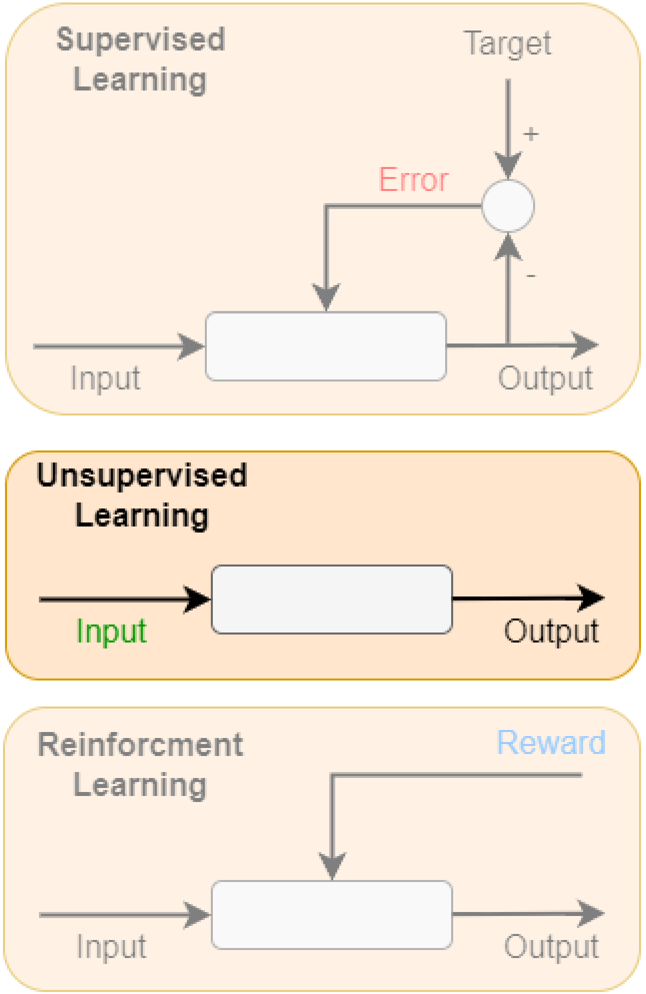



## Supervised Learning

- system is presented with the labeled data
- the objective is to **generalize** the knowledge so that new unlabeled data can be labeled

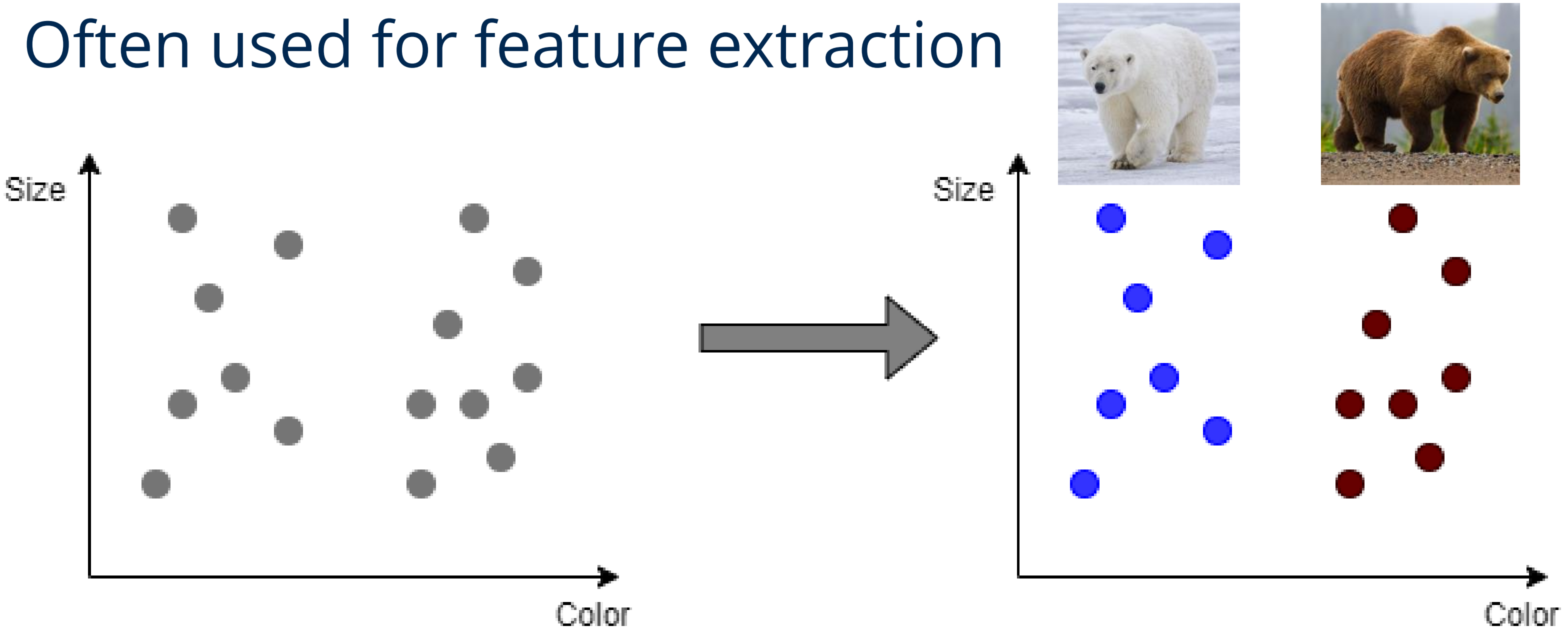


# Learning methods

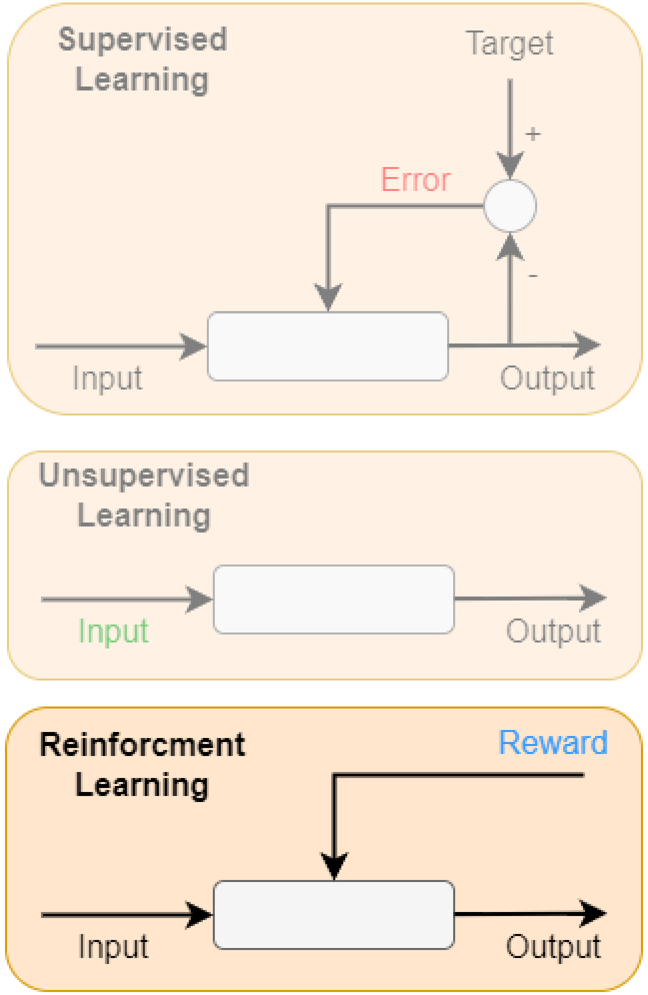



## Unsupervised Learning

- No labels, only has the inputs
- The system uses this data to **learn the hidden structure** of the data so that it can cluster/categorize the data into some broad categories.
- Often used for feature extraction



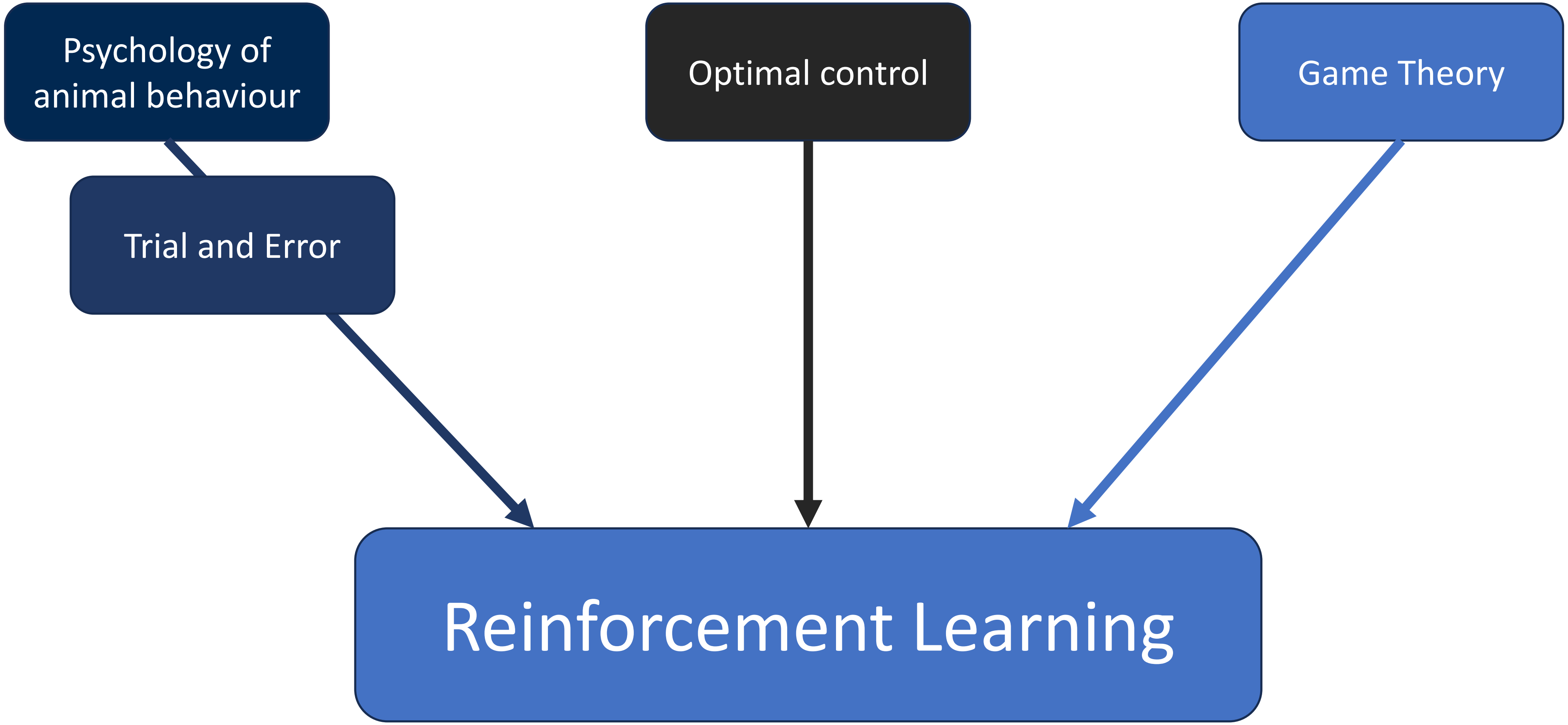
# Learning methods




**Reinforcement Learning**

- The agent does not have prior knowledge of the system. It has to **learn from experience**.
- It gathers feedback and uses that feedback to plan/learn actions to **maximize a specific objective**.
- As it does not have enough information about the environment initially, it must **explore** to gather insights.
- Once it gathers "enough" knowledge, it needs to **exploit** that **knowledge** to start **adjusting** its **behaviour** to **maximize the objective** it is chasing.

# Background


Psychology of animal behaviour
Trial and Error
Optimal control
Game Theory
Reinforcement Learning

# History of RL – Optimal Control

- 1950 - Richard Bellman designed a controller to minimize a measure of a dynamical system's behaviour over time.
- **Bellman equation**
- **Dynamic programming**: class of methods for solving optimal control problems by solving Bellman equation

# History of RL – Optimal Control

- 1950 - Richard Bellman designed a controller to minimize a measure of a dynamical system's behaviour over time.
- **Bellman equation**
- **Dynamic programming**: class of methods for solving optimal control problems by solving Bellman equation
- Problems with Dynamic programming
  - Suffers from „**the curse of dimensionality**"
    - The computational requirements grow exponentially with the number of state variables
  - Computation proceeds backwards in time
    - Needs to be used in forward direction

# History of RL – Optimal Control

- 1950 - Richard Bellman designed a controller to minimize a measure of a dynamical system's behaviour over time.
- **Bellman equation**
- **Dynamic programming**: class of methods for solving optimal control problems by solving Bellman equation
- 1989 – Chris Watkins used **Markov Decision Process (MDP)** formalism in RL tasks

# History of RL – Animal Behaviour

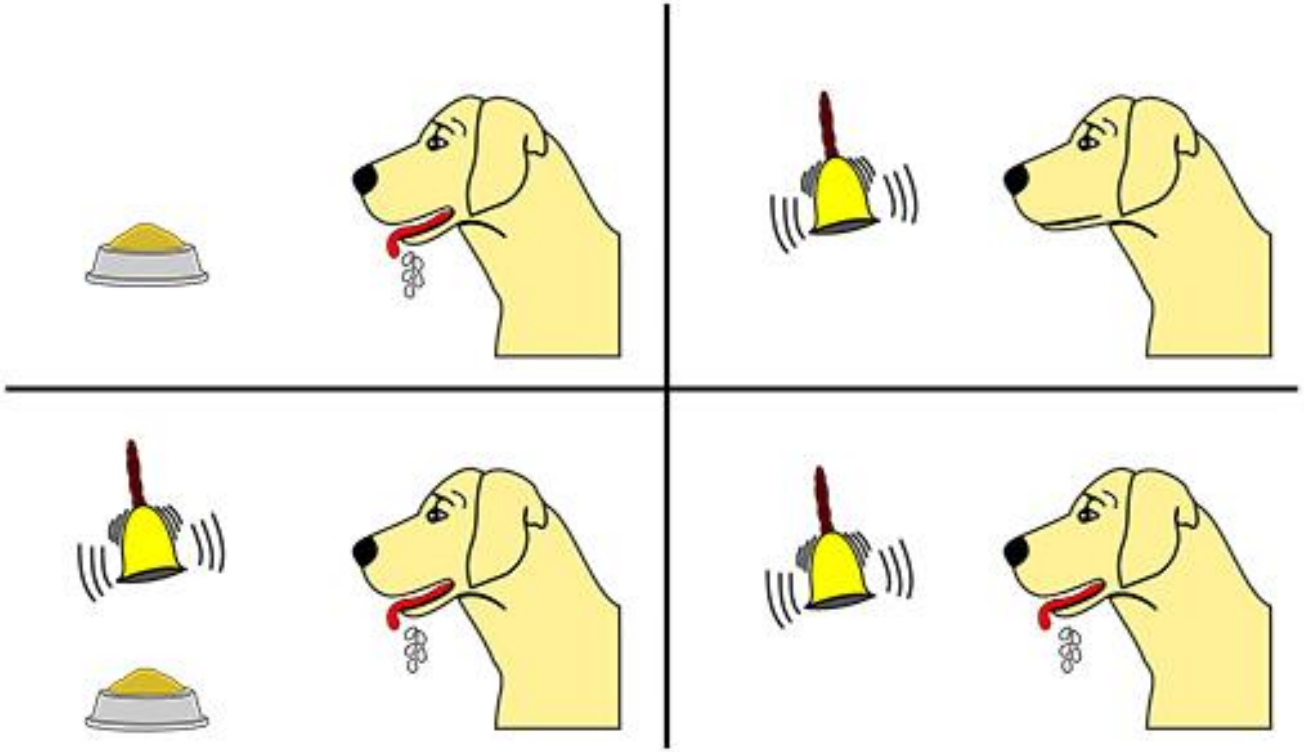
- 1911 – Edward Thorndike – Law of Effect

## Law of Effect:

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.**

# History of RL – Animal Behaviour

- 1911 – Edward Thorndike – Law of Effect
- 1927 – Pavlov – Conditioned reflexes



Pavlov’s dog

# History of RL – Animal Behaviour

- 1911 – Edward Thorndike – Law of Effect
- 1927 – Pavlov – Conditioned reflexes

**Reinforcement in animal learning:**

Reinforcement is the strengthening of a pattern of behavior as a result of an animal receiving a stimulus - a reinforcer - in an appropriate temporal relationship with another stimulus or with a response.

(can be extended with weakening as well)

Reinforcement produces changes in behaviour that persist after the reinforcer is withdrawn.

# History of RL – Animal Behaviour

- 1911 – Edward Thorndike – Law of Effect
- 1927 – Pavlov – Conditioned reflexes
- 1930’ – Implementing Trial and Error learning on machines

**Trial-and-Error learning:**

Selecting actions on the basis of evaluative feedback that does not rely on knowledge of what the correct action should be.

# History of RL – Animal Behaviour

- 1911 – Edward Thorndike – Law of Effect
- 1927 – Pavlov – Conditioned reflexes
- 1930' – Implementing Trial and Error learning on machines
- 1933 – Thomas Ross – Maze solving machine
- 1948 – Alan Turing – Pleasure-pain system
- 1960 – **Reinforcement Learning** as a term were used
- 1961 – Minsky – Basic credit assignment problem for complex reinforcement learning systems

# History of RL

- 1961 – Minsky – Basic credit assignment problem for complex reinforcement learning systems
  **How do you distribute credit for success among the many decisions that may have been involved in producing it?**

# History of RL

- 1961 – Minsky – Basic credit assignment problem for complex reinforcement learning systems
  **How do you distribute credit for success among the many decisions that may have been involved in producing it?**

**Learning automata:**

Methods for solving a nonassociative, purely selectional learning problem known as the **k-armed bandit**.

Simple, low memory machines for improving the probability of reward in these problems.

# History of RL

- 1961 – Minsky – Basic credit assignment problem for complex reinforcement learning systems
  **How do you distribute credit for success among the many decisions that may have been involved in producing it?**
- 1977 – Tabular (TD) methods
- 1989 – Q-learning
- 1992 - Gerry Tesauro - TD-Gammon

  *In 1998, during a 100-game series, it was defeated by the world champion by a mere margin of 8 points.*

# Tabular Solution Methods

- Core ideas of RL
- State and action spaces are small enough for the approximate value functions to be represented as arrays or **tables**
- Often find exact solutions
- Often exactly the optimal value function
- Optimal policy

**Bandit problem:**

*Reinforcement learning problem in which there is only a single state*

# Solving finite MDP

**Dynamic Programming (DP)**

- Well-developed mathematically
- Require a complete and accurate model of the environment

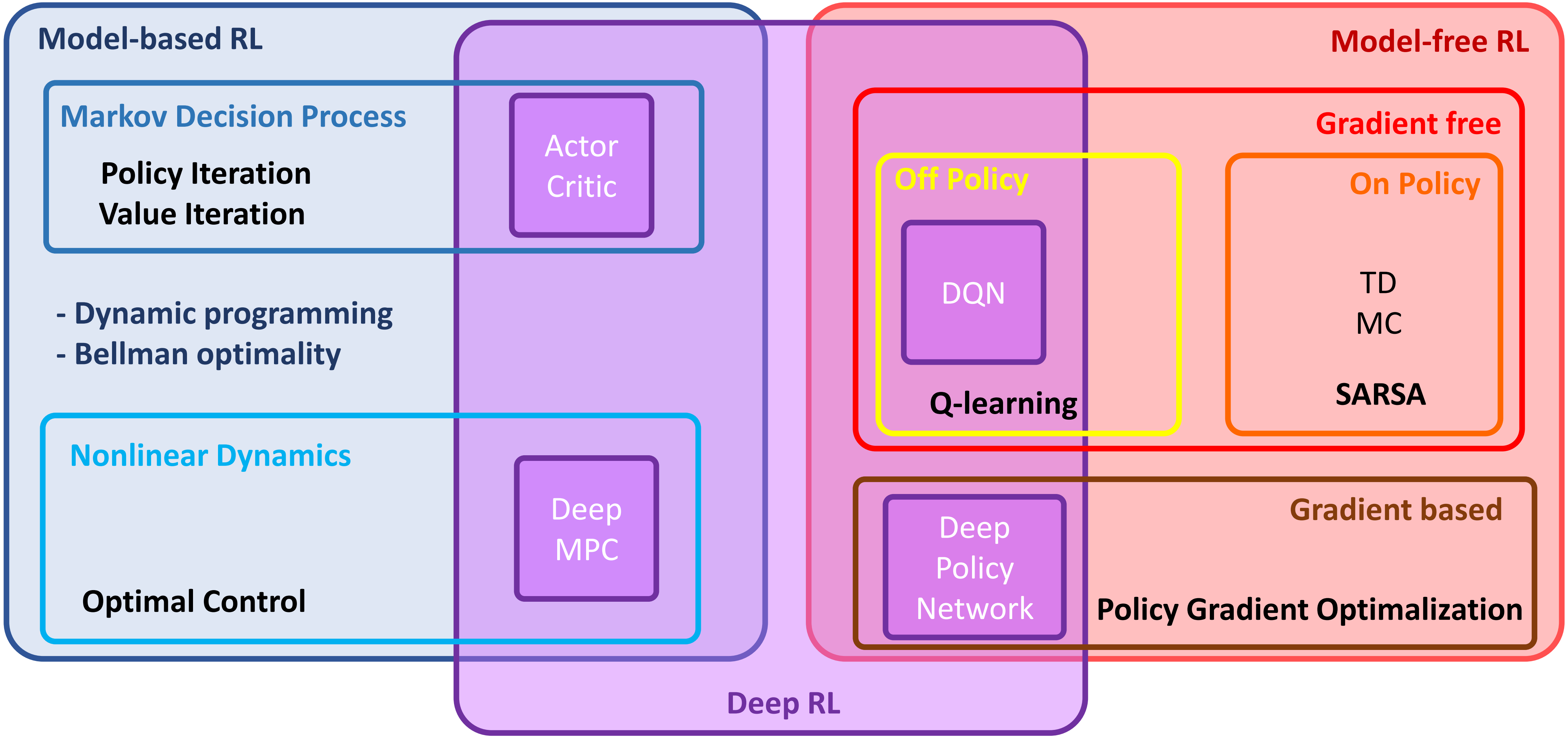
**Monte-Carlo (MC)**

- Do not require a model
- Conceptually simple
- Not well suited for step-by-step incremental computation

**Temporal Difference (TD)**

- Require no model
- Fully incremental
- More complex to analyse

# RL landscape



# Reinforcement Learning (RL) – Key concept

**Explicitly consider the whole problem of a goal-directed agent interacting with an uncertain environment**

# Reinforcement Learning (RL) – Key concept

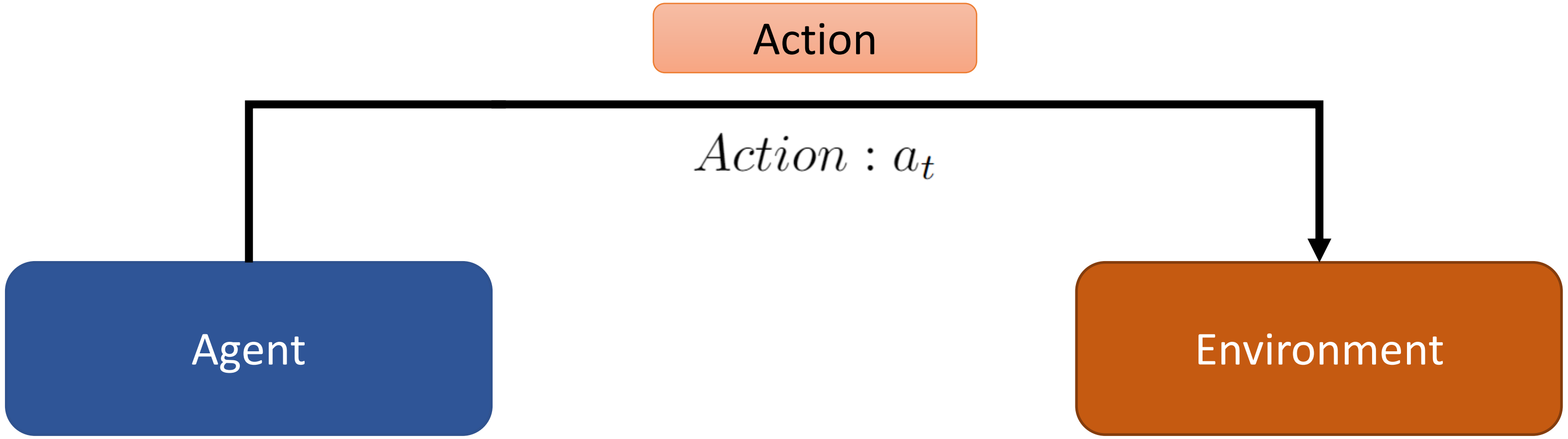
Agent

**Agent:** takes actions

# Reinforcement Learning (RL) – Key concept


**Environment:** the world in which the agent exists and operates

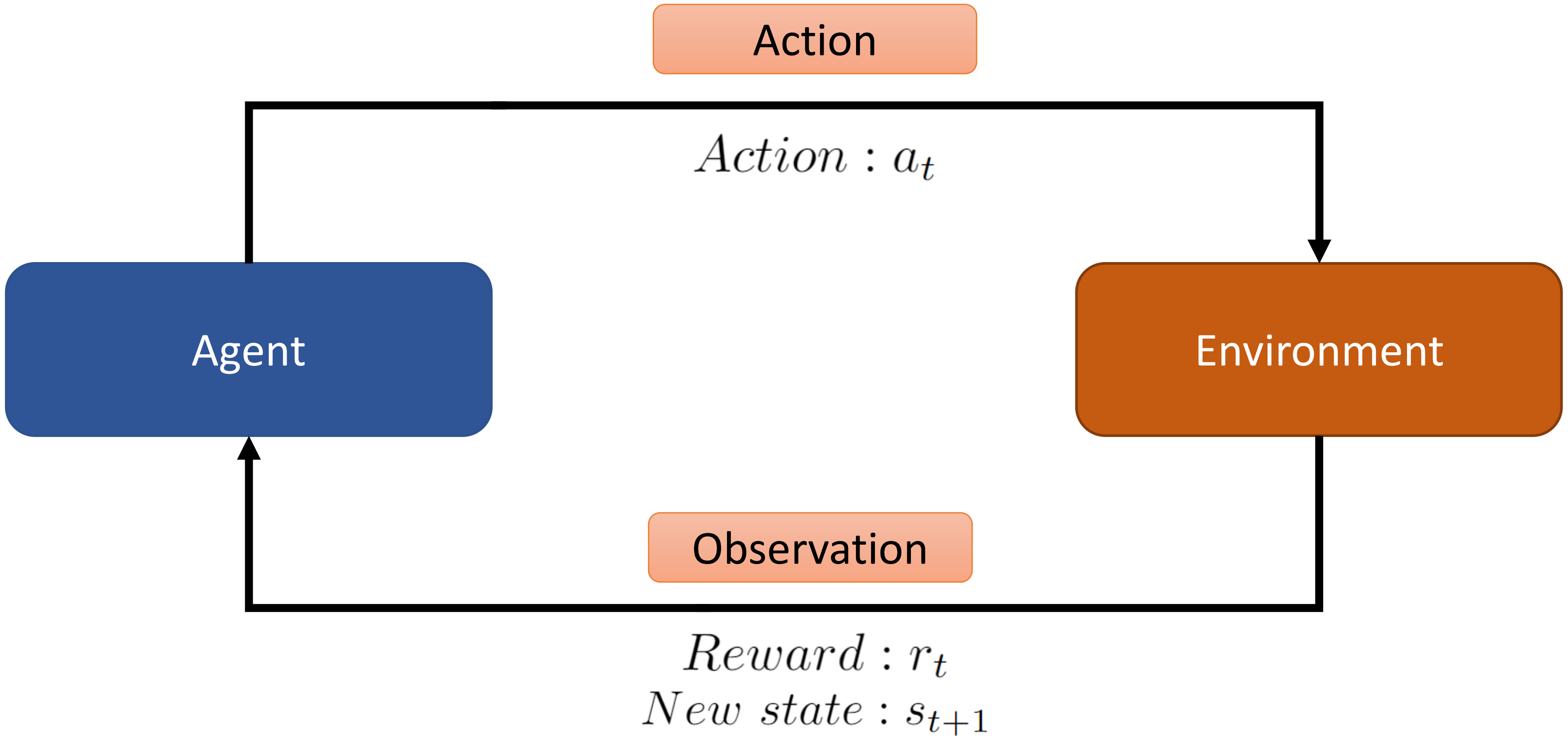
# Reinforcement Learning (RL) – Key concept




**Action:** a movement the agent can make in the environment

**Action space *A***: the set of possible actions an agent can make in the environment

# Reinforcement Learning (RL) – Key concept




**Reward:** feedback that measures the success or failure of the agent's action

# Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward
- Value Function
- Model of the Environment

# Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward
- Value Function
- Model of the Environment

**Agent**:

- Have explicit goals
- Can sense aspects of the environment
- Can choose actions to influence the environment
- Operates despite significant uncertainty about the environment

**Complete, interactive, goal-seeking agent can be:**

- Complete organism
- Robot
- Algorithm
- A component of a larger behaving system
  (Battery charge level monitoring agent)

# Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward
- Value Function
- Model of the Environment

**Environment**:

- In which the agent operates

**State:** a representation of the current environment that the agent is in. This state can be observed by the agent, and it includes all relevant information about the environment that the agent needs to know in order to make a decision

# Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward
- Value Function
- Model of the Environment

**Policy**:

- Learning agent way of behaving
- Determines behaviour
- Stochastic in general

**Stochastic:** refers to the property of being well-described by a random probability distribution

**Deterministic**: no randomness is involved in the development of future states of the system

# Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward
- Value Function
- Model of the Environment

**Reward**:

- Defines the goal in the RL problem
- A single number provided by the environment in each step
- <u>Immediate</u>
- Primary basis for altering the policy

The agent only objective is to maximize the total reward it receives over the long run!

# Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward
- Value Function
- Model of the Environment

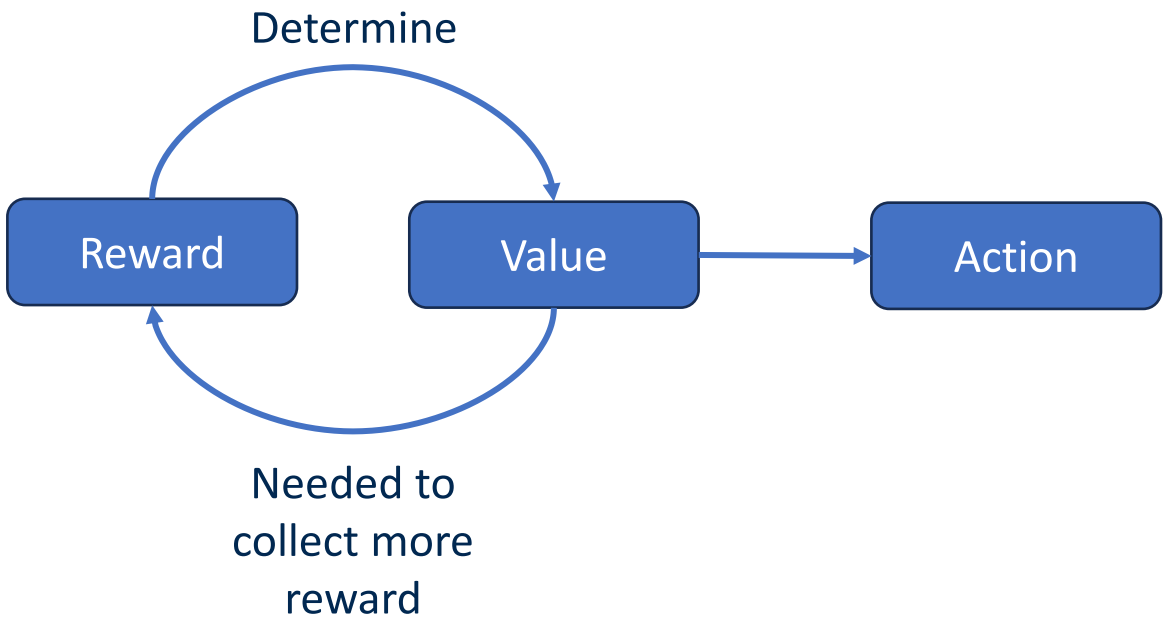
**Value Function**:

- Specifies what is good in the long run
- Action choices made based on value judgement
- Estimated and re-estimated over the lifetime

The **value of a state** is the total amount of reward an agent can expect to accumulate over the future, starting from that state.

# Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward
- Value Function
- Model of the Environment




**Harder to determine value than it is to determine rewards**

# Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward
- **Value Function**
- Model of the Environment

Immediate reward for scoring: 6 points
Immediate reward for surrender: 0 points


Chiefs vs Eagles (Super Bowl 2023)
RB McKinnon

# Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward
- Value Function
- Model of the Environment

**Model of the Environment**:

- The behaviour of the environment
- Model-based RL
  - Models are used for planning
- Model-free RL
  - Models not always given

# Framework for RL

- **Markov Decision Process (MDP)**
  In mathematics, a Markov decision process (MDP) is a discrete-time stochastic control process.

  It provides a mathematical framework for modelling decision making in situations where outcomes are partly random and partly under the control of a decision maker

# Framework for RL

- **Markov Decision Process (MDP)**
  In mathematics, a Markov decision process (MDP) is a discrete-time stochastic control process.

  It provides a mathematical framework for modelling decision making in situations where outcomes are partly random and partly under the control of a decision maker

The **Markov property** expresses that the likelihood of changing to a specific state is reliant exclusively on the present state and elapsed time and not on the series of states that have preceded it

# Elements of MDP

- Environment
- Agent
- State
- Action
- Reward

- *Model*
- *Policy*

# Elements of MDP

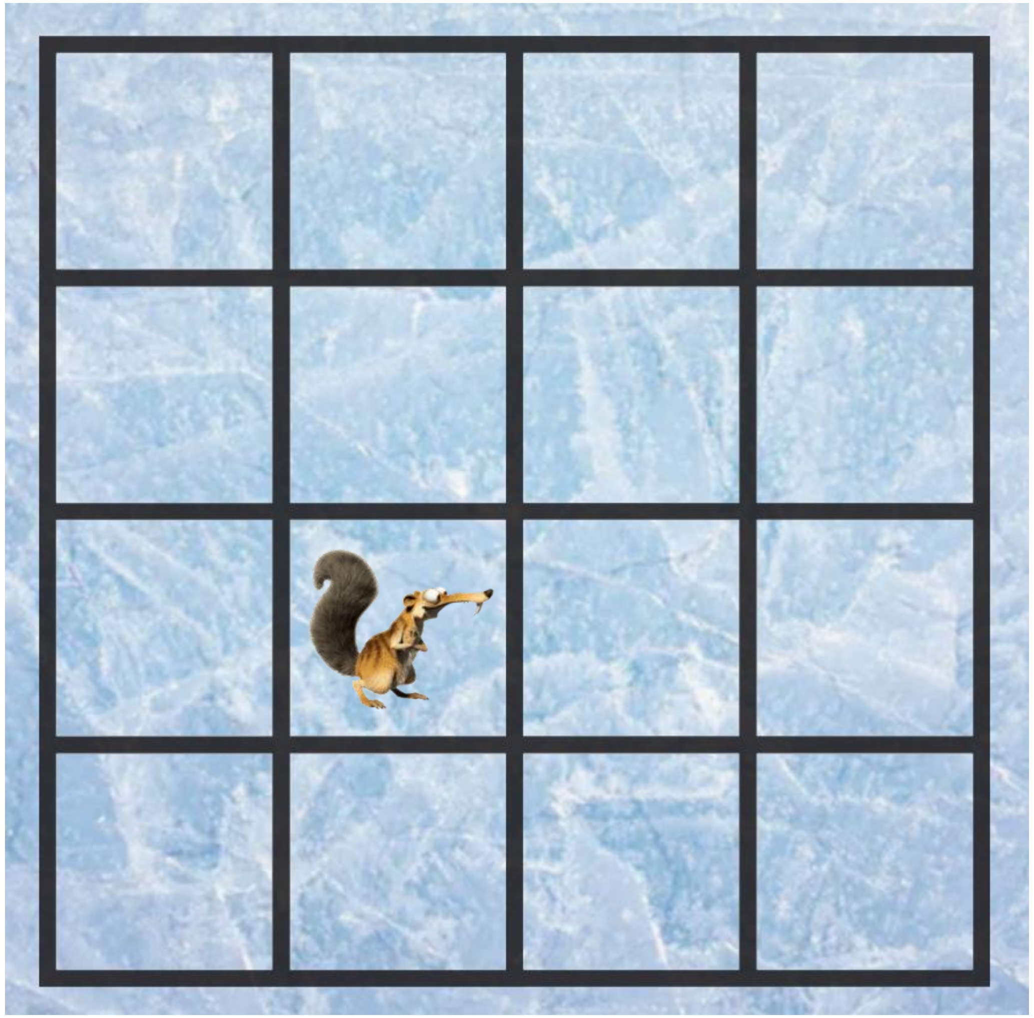
- Environment
- Agent
- State
- Action
- Reward

---

- *Model*
- *Policy*

# Elements of MDP

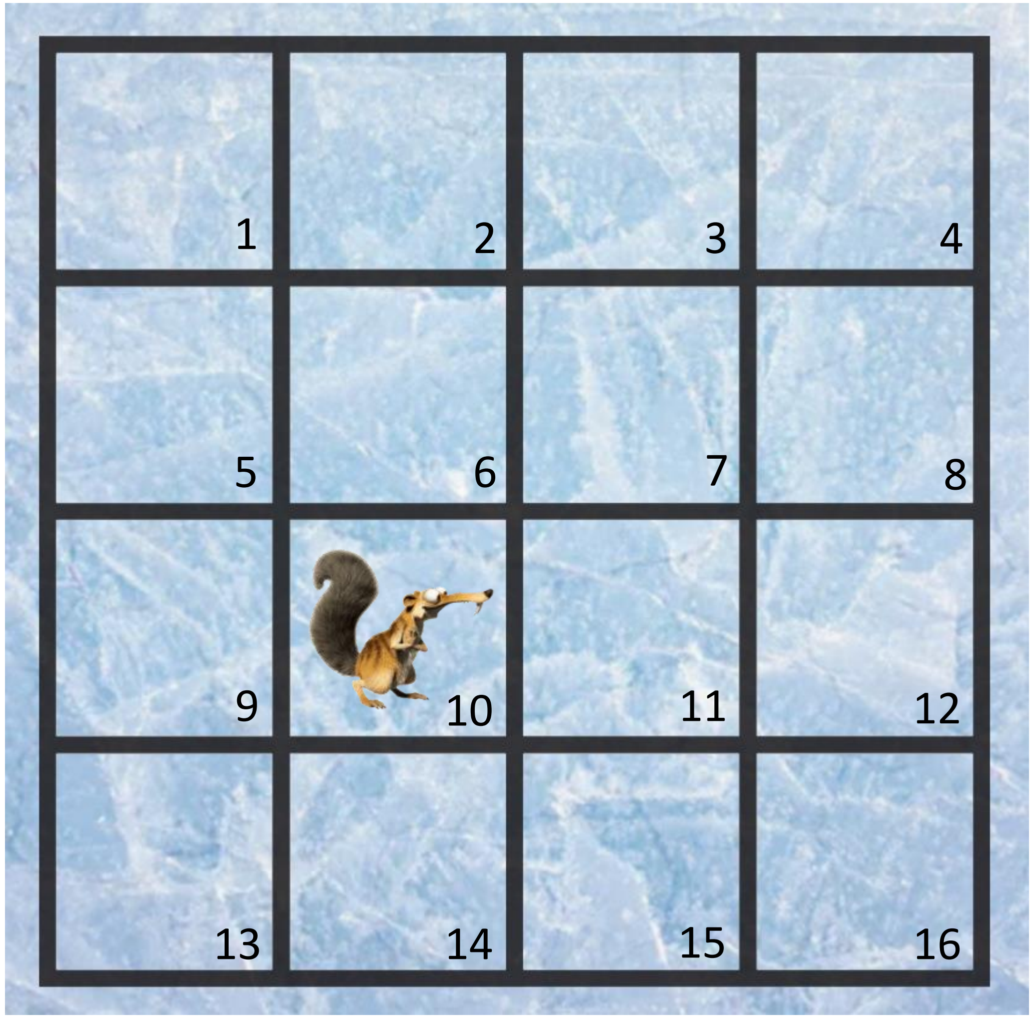
- Environment
- Agent
- State
- Action
- Reward
- *Model*
- *Policy*



# Elements of MDP

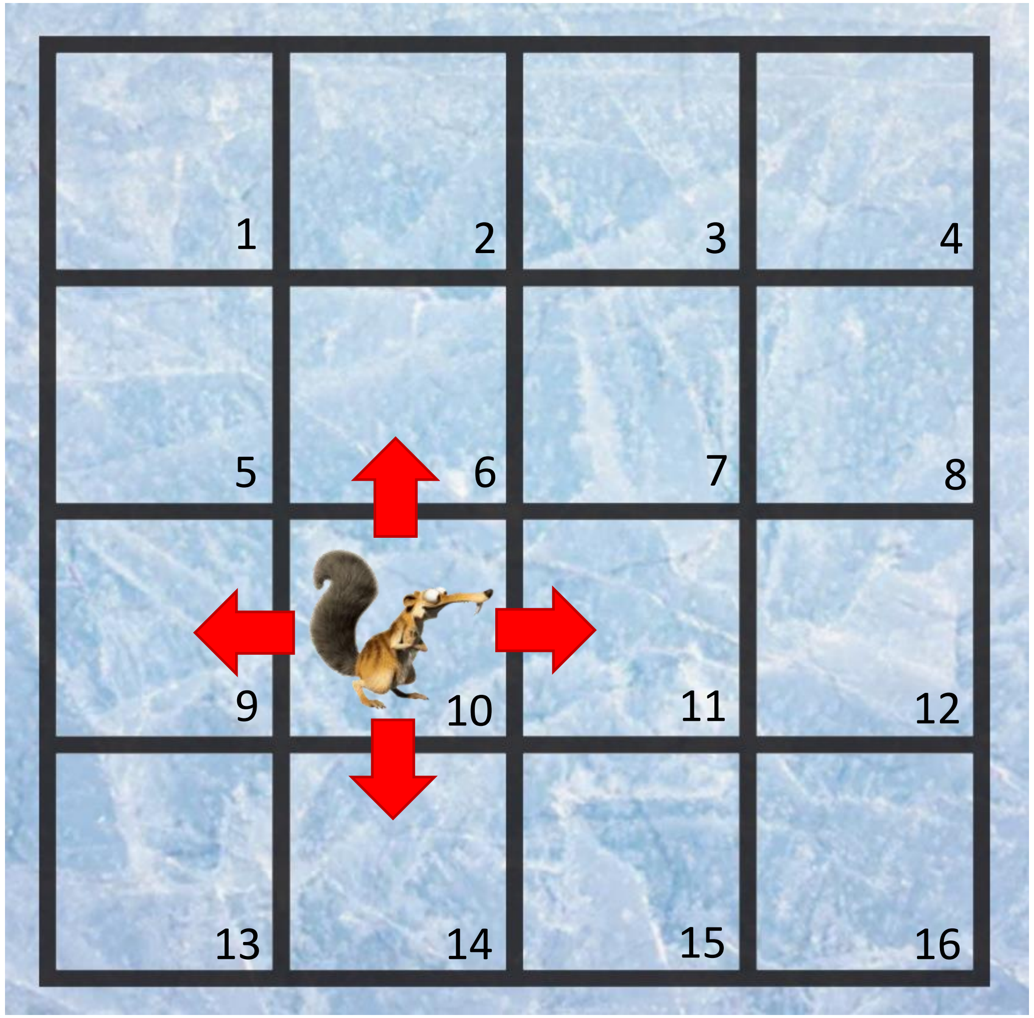
- Environment
- Agent
- State
- Action
- Reward

- *Model*
- *Policy*



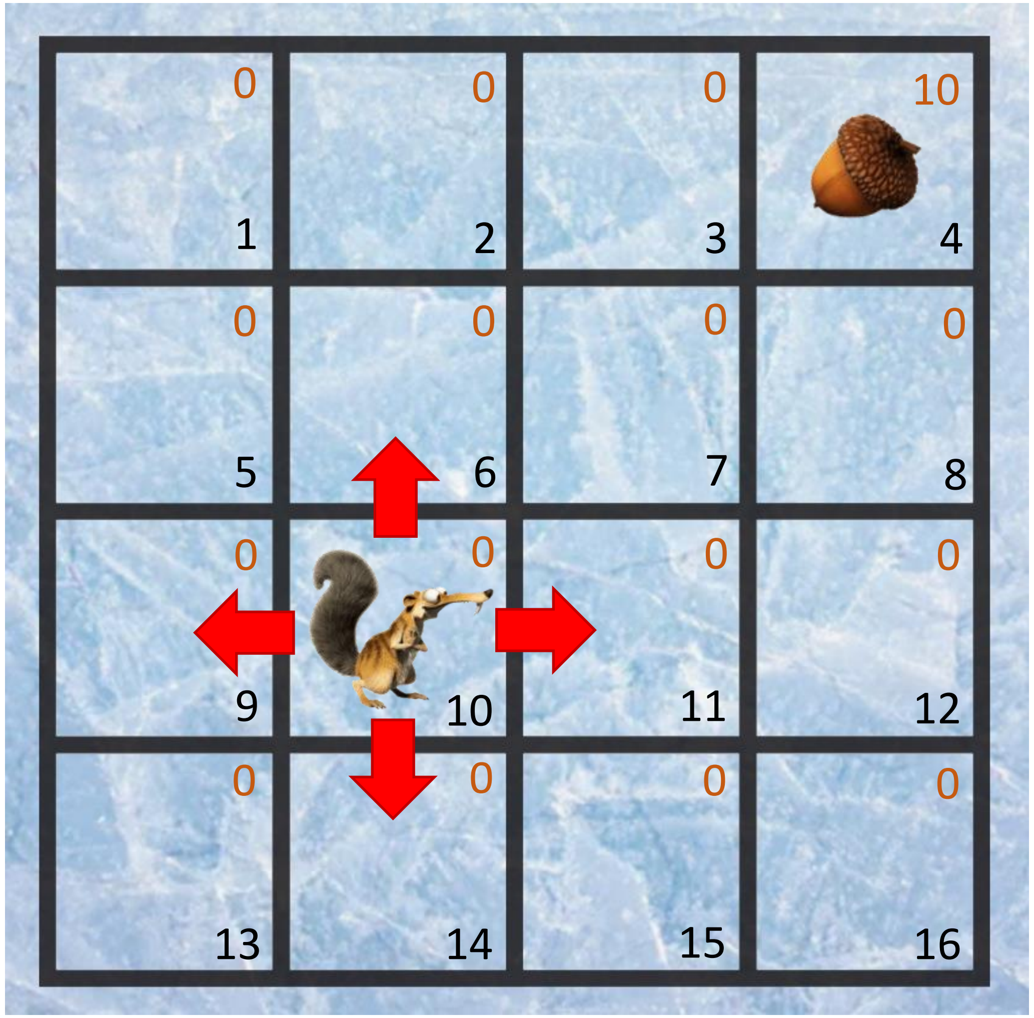
# Elements of MDP

- Environment
- Agent
- State
- Action
- Reward
- *Model*
- *Policy*



# Elements of MDP

- Environment
- Agent
- State
- Action
- Reward
- *Model*
- *Policy*

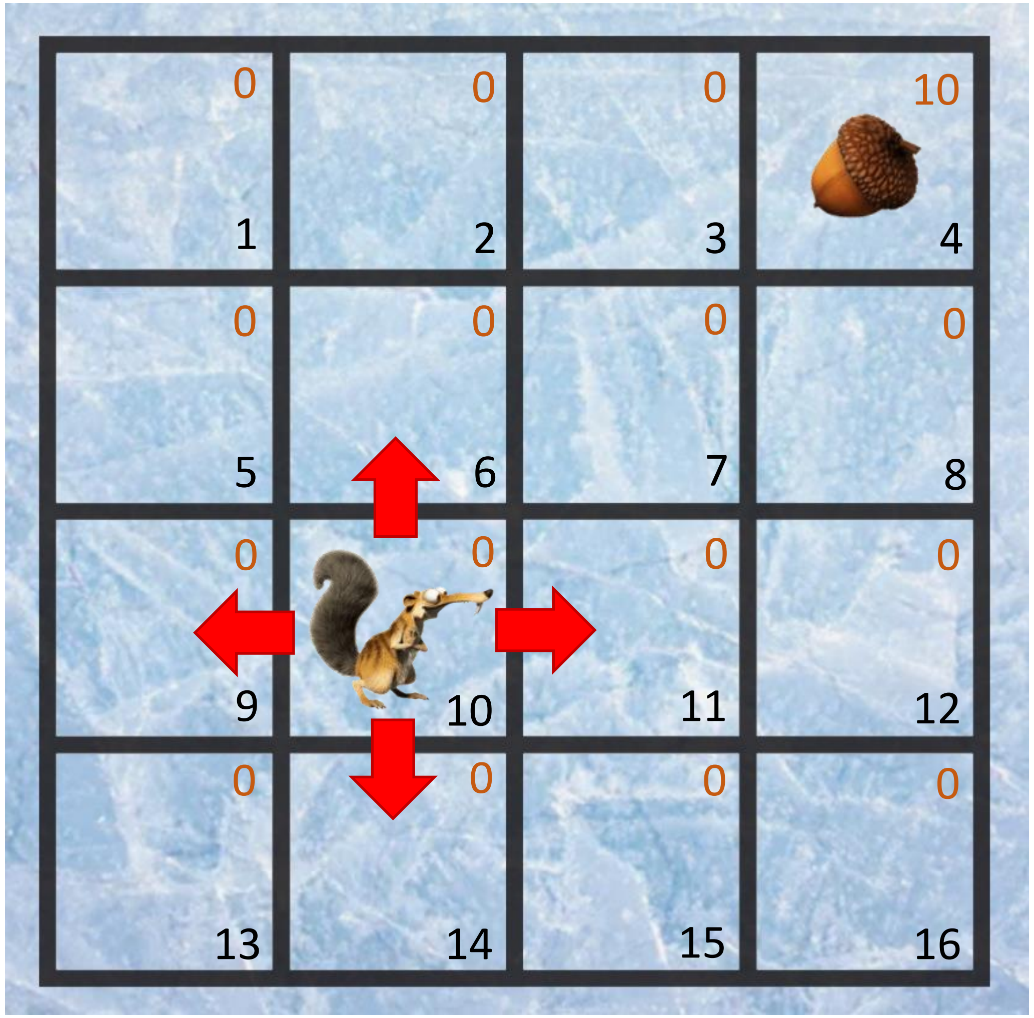


# Elements of MDP

- Environment
- Agent
- State
- Action
- Reward

---

- *Model*
  *rules of the game, the physics of the world*
- *Policy:*
  *A policy is a strategy that an agent uses in pursuit of goals.*



ELTE | FACULTY OF INFORMATICS

# Thank you for your attention!