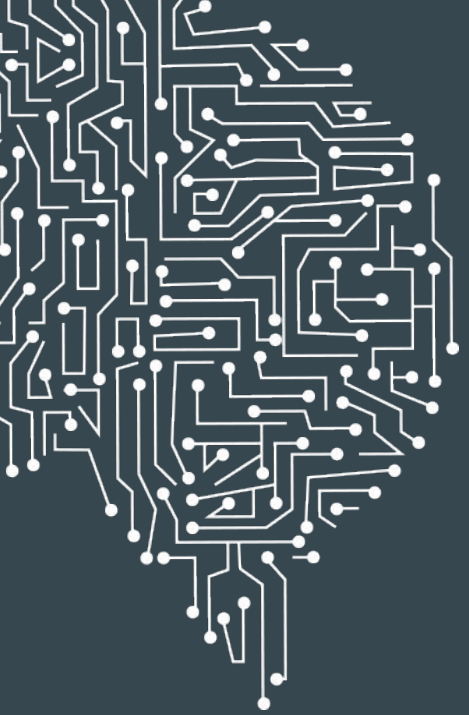


Tutorial 1

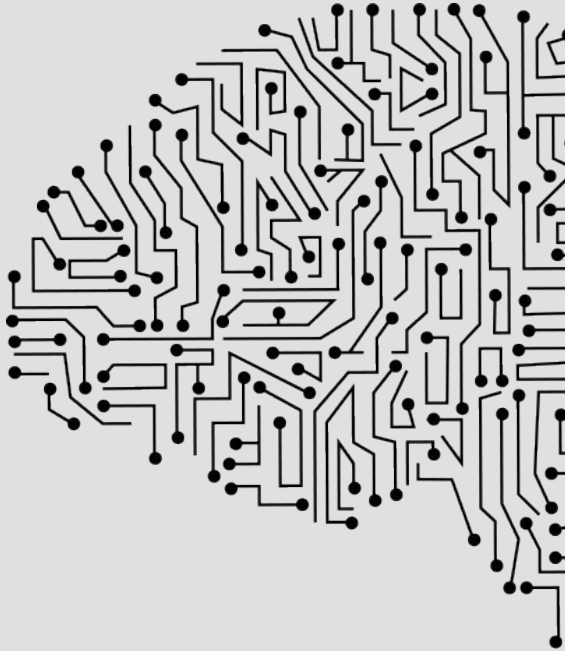
Introducing Artificial Intelligence

Learning Objectives

- General Problems in AI
  - Human actions/thinking
- Strong versus (Weak versus) Narrow AI
- Intelligent Agents
  - Framework for Intelligent Agents
  - Agent Function / Program
- Rational Agents and Performance Measures
- Describing Task Environments (PEAS)
  - Properties of Task Environments
- Types of Agents

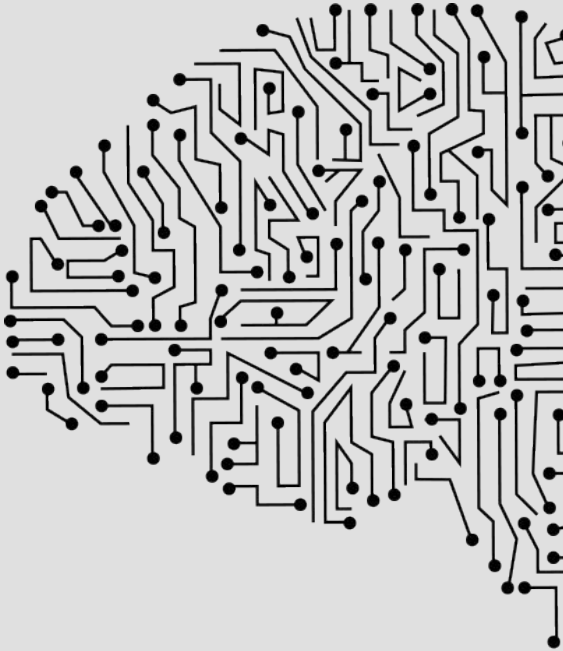
Questions?

https://forms.gle/ZGSmSa2YUMrhGr6e8



Kahoot!

Join at kahoot.it or with the Kahoot! app



Key points Revisited

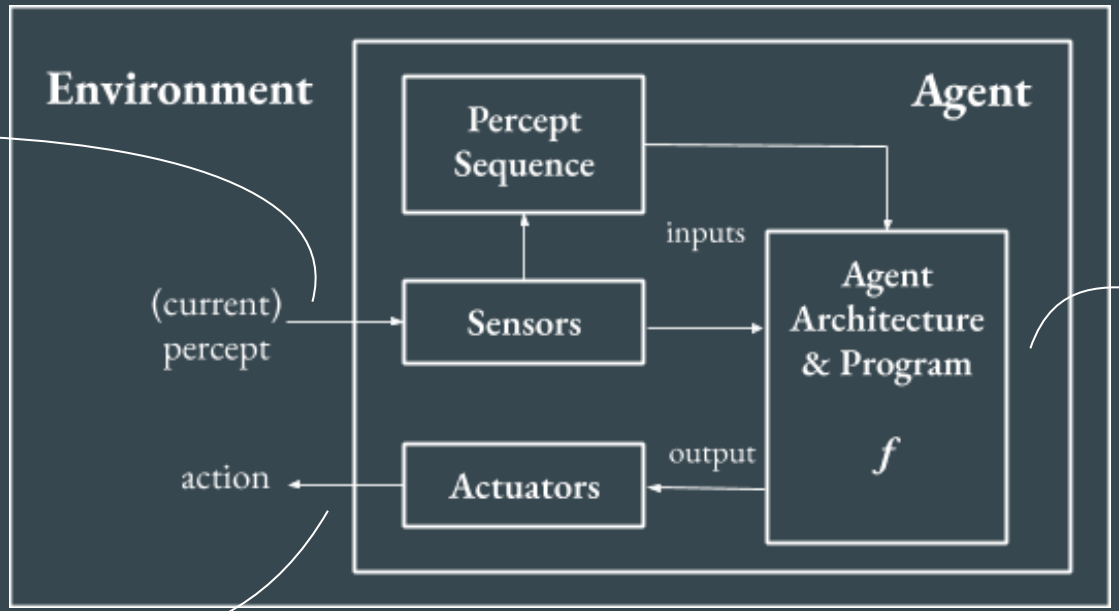
- Agent
  - = AI mechanism (in our case, the agent function)
  - More generically, agent = the agent function ***f***, such that ***a = f(s)***
    (s : a collection of inputs; a : one of many possible actions/decisions)

- Rationality
  - Rational mechanism : selects an action that is expected to maximise its performance measure for each possible percept sequence, given the evidence provided by the percept sequence and the agent's built-in knowledge.

Key points Revisited (Cont.)

- Rational Agent Structure

1. Agents collects information about its environment via percepts (i.e., data about its environment)



2. Agent function uses percepts (current, and sometimes also historical) as input data to generate an output (i.e., an action)

3. Agent performs actions through actuators (i.e., mechanisms by which it interacts and changes its environment)

Key points Revisited (Cont.)

- PEAS task environment specification
  - Performance measure, Environment, Actuators, Sensors
- Other environmental properties
  - Fully vs. partially observable
  - Deterministic vs. stochastic
  - Static vs. dynamic
  - Episodic vs. sequential
  - Discrete vs. continuous
  - Known vs. unknown
  - Single vs. multiagent

helps determine →

what **data** is obtainable,

what **representations** may be utilised,

difficulty of the problem

Key points Revisited (Cont.)

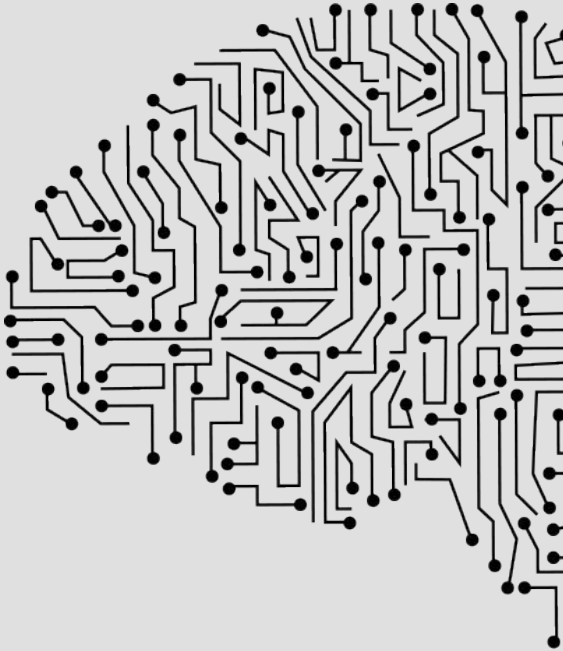
- Type of agents
  - Reflex agent
  - Model-based (reflex) agent
  - Goal-based agent
  - Utility-based agent

helps determine →

representation (i.e., form/structure) of ***f***

- 4 elements of learning agents
  - Learning element
  - Performance element
  - Critic
  - Problem generator

Applied Questions



Tic-Tac-Toe vs Hanabi

https://www.ultraboardgames.com/hanabi/game-rules.php

Tic-Tac-Toe vs Hanabi

Environment characteristic	Tic-Tac-Toe	Hanabi
★ Fully/partially observable?	Fully observable	Partially observable
Single/multi-agent?	Multi-agent	Multi-agent
★ Deterministic/stochastic?	Deterministic	Stochastic
★ Episodic/sequential?	Sequential	Sequential
Static/dynamic?	Static	Static
Discrete/continuous?	Discrete	Discrete

Tic-Tac-Toe vs Hanabi (1)

- **Fully or partially observable?**

Tic-Tac-Toe

- Each agent can see everything the opponent has done and is doing
- All aspects of the task environment that are *relevant* can be detected by sensors
- Agents need not maintain internal state

⇒ Fully observable

Hanabi

- Agents cannot see their own cards → Do not have complete knowledge of all the cards that can be played (i.e. available actions at each state)
- Agents cannot see the cards in the deck

⇒ Partially observable

Tic-Tac-Toe vs Hanabi (2)

- **Single or multi-agent?**

Tic-Tac-Toe

- Has 2 players
- Opponent's performance measure depends on other player's behaviour (*competitive*)

Hanabi

- For 2-5 players
- Players play together as a team to achieve a common objective (*cooperative*) → players need to *communicate*

⇒ Both are multi-agent

Tic-Tac-Toe vs Hanabi (3)

- **Deterministic or stochastic?**

more about states in later topics!

Tic-Tac-Toe

- Next state is *completely determined* by current state + action taken by agent
- Even though each agent cannot predict opponent's action with certainty

⇒ Deterministic

Hanabi

- Drawing cards from a randomised deck introduces uncertainty (wrt the available actions in a state)
- Taking the same action (drawing a card) from the same initial state does not result in the same resultant state with certainty

⇒ Stochastic

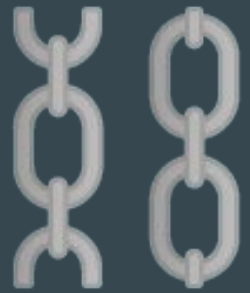
Tic-Tac-Toe vs Hanabi (4)

- **Episodic or sequential?**

Tic-Tac-Toe | **Hanabi**

- Current decision can affect all future decisions (agent needs to think ahead)
- Current decision depends on all past decisions
- If we define a *series of games* (e.g. a tournament), then each game can be an *episode*
  - ➢ Each game can be considered independently

⇒ Both are sequential



Tic-Tac-Toe vs Hanabi (4)

- **Episodic or sequential?**

Tic-Tac-Toe

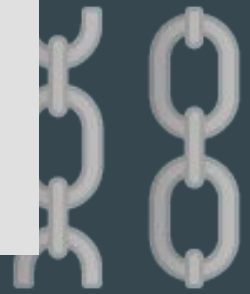
- ❖ Current d
- ❖ Current d
- ❖ If we defi
  - ➢ Each

⇒ Both are seq

Episodic:

- Agent's experience is divided into independent episodes; agent receives a percept and performs a single action in each episode
- Next episode does not depend on previous, e.g. classifying a series of objects on a conveyor belt
- Current decisions do not affect future states

episode



Tic-Tac-Toe vs Hanabi (5)

- **Static or dynamic?**

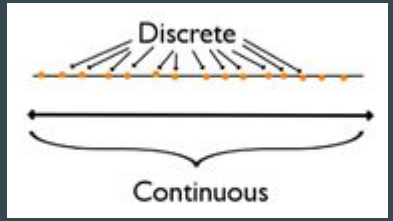
Tic-Tac-Toe | **Hanabi**

- Task environment does not change *while agent is deliberating*
  - Is the environment continuously asking the agent what it wants to do?
  - Does taking time to decide on an action count as deciding to do nothing?
- Agents need not keep looking at the environment while deciding
- Agents need not worry about the passage of time
  - There are also *semi-dynamic* environments: environment itself does not change with time but agent's performance score does

⇒ Both are static

Tic-Tac-Toe vs Hanabi (6)

- **Discrete or continuous?**



Tic-Tac-Toe

- Finite, discrete set of actions that can be taken (i.e. play an X or O at one of the 9 spaces that is not already occupied)
- Environment state is discrete
- Turn-based (or discrete time-steps)

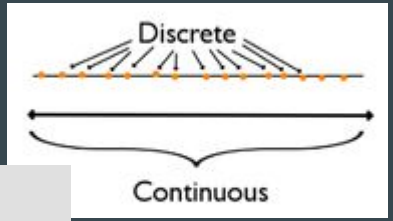
Hanabi

- Finite, discrete set of actions that can be taken (give information, discard a card, play a card)
- Environment state is discrete
- Turn-based (or discrete time-steps)

⇒ Both are discrete

Tic-Tac-Toe vs Hanabi (6)

- **Discrete or continuous?**



Tic-Tac-Toe

- ❖ Finite, discre… actions that can be taken (i.e. pla… on, discard a card, spaces that is…
- ❖ Environment… discrete
- ❖ Turn-based (… -based)

⇒ Both are discret…

Examples of continuous state spaces:

- Space for an autonomous vehicle
- Money (where we are not restricted by cents)

(Do not need to concern ourselves with non-discrete time-steps)

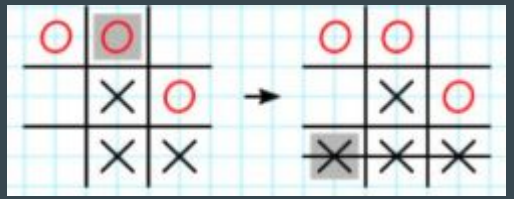
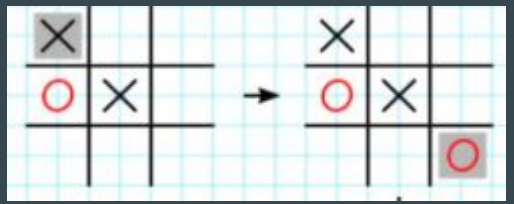
→ Can divide continuous state spaces into discrete ones

Tic-Tac-Toe Reflex Agent

- **Reflex agent**
  - Uses condition-action rules
    - Matches “cases” in lookup table
    - Does not try to predict
  - Action is selected based *only* on the *current percept*
    - Percept history is ignored

Note: tic-tac-toe grid is symmetric by flipping and rotation, so the number of configurations is relatively small :)

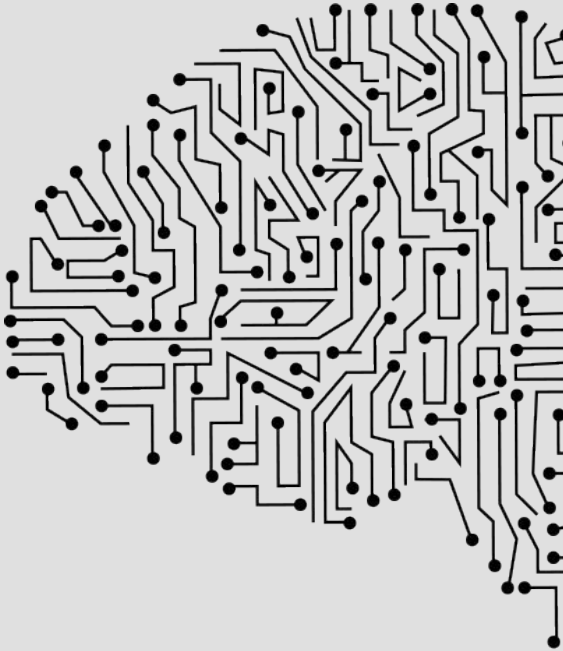
Tic-Tac-Toe Reflex Agent (1)

Priority	Percepts	Action
1	❖ Win! Player has 2 in a row, and the remaining cell is not blocked.	Place the 3rd. 
2	❖ Block the opponent Opponent has 2 in a row, and the remaining cell is not blocked.	Place the 3rd. 

Refer to https://en.wikipedia.org/wiki/Tic-tac-toe#Strategy for more!

Questions?

https://forms.gle/ZGSmSa2YUMrhGr6e8



Thank you!

...