§1: Introduction, Agents

CS 325 - Artificial Intelligence

Note

1. No Required Text book.
2. Slides and other resources will be provided.
3. The slides and other resources will be sufficient for the assignments and the exams.
4. Lecture is based on the slides and demos.
5. Attending the class is HIGHLY recommended.

Question

Definition

What is Artificial Intelligence?

Question

Definition

What is Artificial Intelligence?
What is the objective of designing an AI?

Question

Definition

What is Artificial Intelligence?
What is the objective of designing an AI?
What is Intelligence?

Question

Definition

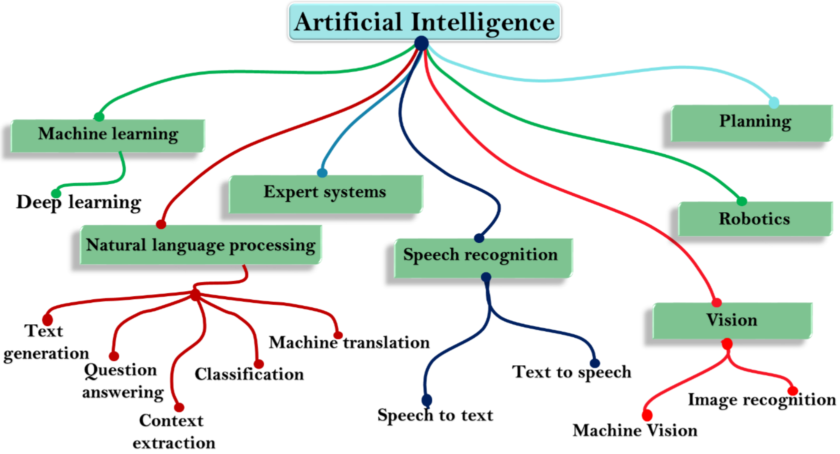
What is Artificial Intelligence?
What is the objective of designing an AI?
What is Intelligence?
Intelligence: capacity of Learning, Reasoning, Understanding, and Problem Solving.
One simple goal: an AI should make decisions.

Definition

Definition

Artificial Intelligence?
Building a program or machine that appears to be intelligent.
AI is the science that making computers do things that requires intelligence when done by human.

Subsets of AI

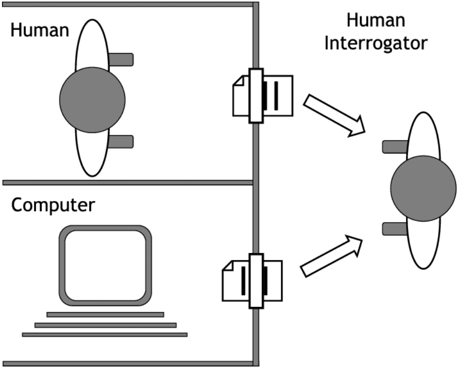


The Turing Test

In 1950 – Alan Turing devised a test for intelligence called the Imitation Game.

Ask questions of two entities, receive answers from both If you can't tell which of the entities is human and which is a computer program, then you are fooled and we should therefore consider the computer to be intelligent.

The Turing Test

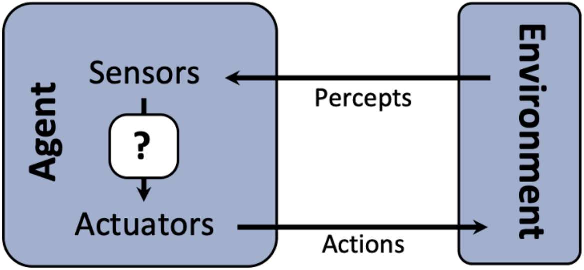


AI in Real World

Planning and Scheduling
Self-Driving Cars
Pattern Recognition
Image Processing / Facial Recognition
Robotics
Gaming

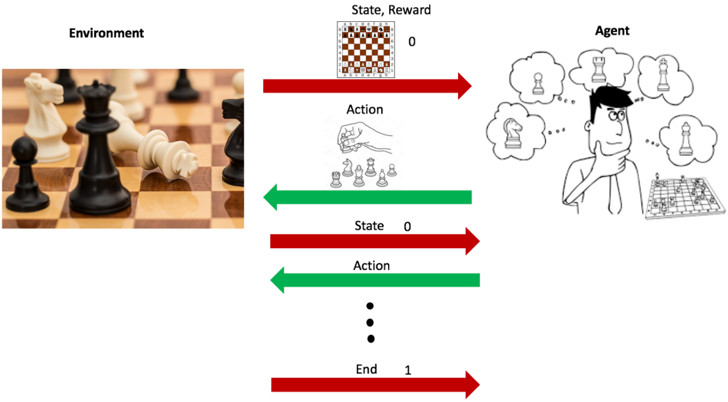
Agent & Environment

An agent is an entity that perceives and acts.
Action affect the environment.
Agent makes decisions and takes actions.
Agent has a goal (solve the puzzle, get the most points).



Agent & Environment

In most cases, environment makes a problem while the agent gives a solution (problem: my room is dirty, solution: Roomba Vacuum Cleaner)

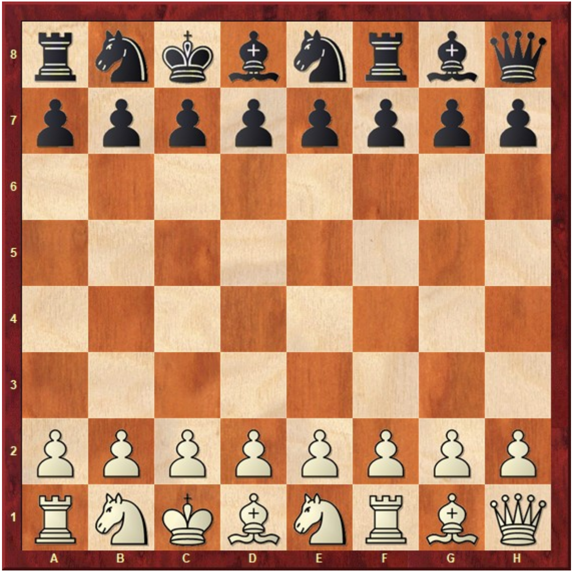


Agent & Environment

The agent takes action(s) which changes that environment and the process repeat based on new environment.
The agent perceives its environment through sensors (camera, laser).
The agent acts on its environment through actuators (arms, wheels).
The agent may have some knowledge about the environment.
AI is the algorithmic way of making those decisions more intelligent.
Environment:

- Rules and Boundaries.
- Legal Actions
- Starting State
- Goal State

Environment: example

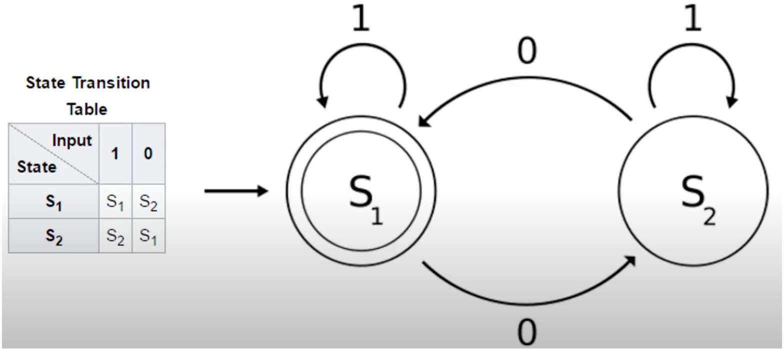


Environment Current State

Agent's perceptual inputs at any instant.
Environment Current State: Agent's current belief of the world.
Agent's choice of action is based on entire precept sequence of data.
An action taken by agent in a given state transitions the state to another state.

State Transition Function:

State Transition Function:
State: S, Action: A
$S' = STF(S, A)$



State Transition Table

State \ Input	1	0
S_1	S_1	S_2
S_2	S_2	S_1

Rationality

A rational agent is one that does the optimal action for every possible state.
Optimal action: the action makes the agent most successful.
How to measure the agent success?
We need performance measure/evaluation function to measure how successful the agent has been.

Environment types

- Fully observable (vs. partially observable)
- Deterministic (vs. stochastic)
- Episodic (vs. sequential)
- Static (vs. dynamic)
- Discrete (vs. continuous)

Environment types

- **Fully observable (vs. partially observable)**
  Is everything an agent requires to choose its actions available to it via its sensors?
  If so, the environment is fully accessible.
  If not, parts of the environment are inaccessible Agent must make informed guesses about world
- Fully observable: agent's sensors describe environment state fully.
- Partially observable: some parts of environment not visible, noisy sensors.
- Example: Video games are partially observable. Chess is fully observable.

Environment types

- **Deterministic (vs. stochastic)**
  Does the change in world state Depend only on current state and agent's action?
  Deterministic environments: the next state of environment is entirely determined by the current state and the action of agent.
  Non-deterministic environments: Have aspects beyond the control of the agent.

Environment types

- **Deterministic (vs. stochastic)**
- Deterministic: next state fully determined by current state and agent's actions.
- Stochastic: random changes (can't be predicted exactly).
  An environment may appear stochastic if it is only partially observable.
  Uncertainty about the outcomes of actions (Dice roll, Card draw).
  Agent doesn't know what coming next (poker game).

Environment types

- **Episodic (vs. sequential)**
  Is the choice of current action Dependent on previous actions?
  If not, then the environment is episodic
  In non-episodic/ sequential environments: Agent has to plan ahead:
  Current choice will affect future actions.
  Action may have a long-term consequences.

Environment types

- **Static (vs. dynamic)**
  Static: Environment only change when agent takes action.
  Dynamic: Environment may change while agent decides.
  Static environments don't change While the agent is deliberating over what to do
  Dynamic environments do change So agent should/could consult the world when choosing actions Alternatively: anticipate the change during deliberation OR make decision very fast

Environment types

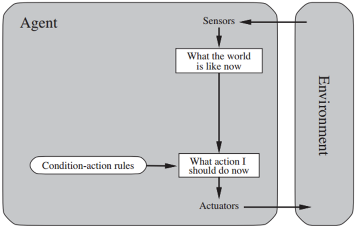
- **Discrete (vs. continuous)**
  Discrete: finite number of distinct states.
  Discrete set of percepts and actions.
  Continuous: continuous time/actions.
  A limited number of distinct, clearly defined percepts and actions or a big range of values (continuous).

Agent Types

- **Simple Reflex Agents:** are based on condition-action rules and implemented with an appropriate production system. They are stateless devices which do not have memory of past world states.
- **Model-Based Reflex Agents:** have internal state which is used to keep track of past states of the world.
- **Goal-Based Agents:** are agents which in addition to state information have a kind of goal information which describes desirable situations. Agents of this kind take future events into consideration.
- **Utility-Based Agents:** use internal estimate for performance measure to compare future states.

Simple Reflex Agents

Simple reflex agents ignore the rest of the percept history and act only on the basis of the current percept.
The agent function is based on the condition-action rule. A condition-action rule is a rule that maps a state i.e, condition to an action. If the condition is true, then the action is taken, else not.
This agent function only succeeds when the environment is fully observable.



Simple Reflex Agents

Here's how a Simple Reflex Agent typically works:

- Perception: It perceives the environment through sensors. These sensors gather information about the current state of the environment.
- Condition: It has a set of pre-defined condition-action rules. These rules are usually in the form of "IF [certain condition is met] THEN [take a specific action]."
- Action: When the agent's sensors detect a condition that matches one of its rules, it immediately takes the corresponding action.

Simple Reflex Agents

- Agent works by finding a rule whose condition matches the current situation (rule-based systems).
- But, this only works if the current percept is sufficient for making the correct decision.
- Example: IF it is dark THEN turn on lights

Simple Reflex Agents

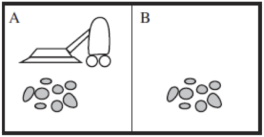


Figure 2.2 A vacuum-cleaner world with just two locations.

Percept sequence	Action
[A, *Clean*]	*Right*
[A, *Dirty*]	*Suck*
[B, *Clean*]	*Left*
[B, *Dirty*]	*Suck*
[A, *Clean*], [A, *Clean*]	*Right*
[A, *Clean*], [A, *Dirty*]	*Suck*
⋮	⋮
[A, *Clean*], [A, *Clean*], [A, *Clean*]	*Right*
[A, *Clean*], [A, *Clean*], [A, *Dirty*]	*Suck*
⋮	⋮

Simple Reflex Agents

- Simple but very limited intelligence
- If there occurs any change in the environment, then the collection of rules need to be updated.
- Infinite loops.
  Suppose vacuum cleaner does not keep track of location. What do you do on clean left of A or right on B is infinite loop Randomize action.

Simple Reflex Agents

These agents are efficient for straightforward tasks in controlled environments but are not suitable for more complex or dynamic scenarios where planning, memory, and learning are required.
More advanced agents like Model-Based Reflex Agents or Goal-Based Agents are used for handling such situations.

Model-Based Reflex Agents

- While Simple Reflex Agents operate based on condition-action rules without any internal representation of the world, Model-Based Reflex Agents incorporate a basic model or representation of the world to make more informed decisions.

Model-Based Reflex Agents

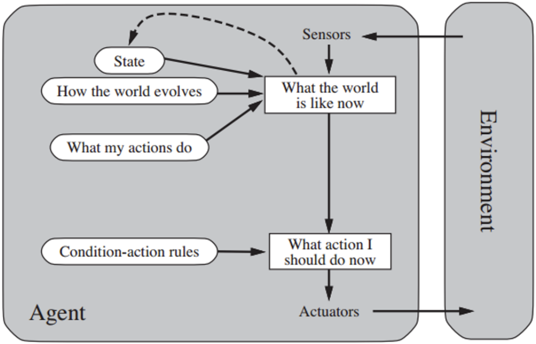
- A model-based agent can handle partially observable environments by the use of a model about the world.
- The agent has to keep track of the internal state which is adjusted by each percept and that depends on the percept history.
- The current state is stored inside the agent which maintains some kind of structure describing the part of the world which cannot be seen.

Model-Based Reflex Agents

Here's how a Model-Based Reflex Agent typically works:

- Perception: it perceives the environment through sensors. These sensors provide information about the current state of the environment.
- Internal Model: A Model-Based Reflex Agent maintains an internal model or representation of the world. This model includes information about the current state of the environment, past actions taken, and the expected outcomes of those actions.
- Condition-Action Rules: It has condition-action rules similar to a Simple Reflex Agent, but it can also take into account the information in its internal model. So, the rules might be more sophisticated, such as "IF [certain condition is met] AND [internal model indicates a certain state] THEN [take a specific action]."
- Action: When the agent's sensors detect a condition that matches one of its rules, it takes the corresponding action. However, it may also consider its internal model to make more context-aware decisions.

Model-Based Reflex Agents



Model-Based Reflex Agents

- Updating internal state requires two kinds of encoded knowledge.
  Knowledge about how the world changes (independent of the agents' actions).
  Knowledge about how the agents' actions affect the world
- But, knowledge of the internal state is not always enough
  How to choose among alternative decision paths (e.g., where should the car go at an intersection)?
  Requires knowledge of the goal to be achieved

Goal-Based Agents

Unlike Simple Reflex and Model-Based Reflex Agents, which primarily react to immediate conditions, Goal-Based Agents have the ability to set goals, plan actions to achieve those goals, and execute those plans.

Goal-Based Agents

These kinds of agents take decisions based on how far they are currently from their goal(description of desirable situations).
Their every action is intended to reduce its distance from the goal. This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state.
The knowledge that supports its decisions is represented explicitly and can be modified, which makes these agents more flexible.
They usually require search and planning. The goal-based agent's behavior can easily be changed.

Goal-Based Agents

Reflex agents only act based on pre-computed knowledge (rules).
Goal-based (planning) agents act by reasoning about which actions achieve the goal.
Knowing about the current state is not always enough to decide what to do.
The agent needs some sort of goal information that describes situations that are desirable.

Goal-Based Agents

Here's how a Goal-Based Agent typically works:

- Goal Setting: The agent starts with one or more predefined goals or objectives.
- Perception: it perceives the environment through sensors, gathering information about the current state of the world.
- Reasoning and Planning: The agent uses its internal knowledge, which can include a model of the environment, to reason about the current state and determine a course of action to achieve its goals. This involves planning, which is the process of selecting a series of actions that will lead to the desired outcome.

Goal-Based Agents

[Continued]

- Action Execution: The agent executes the actions it has planned to move toward its goals. It monitors the environment during action execution to ensure that its plan is still relevant and makes adjustments if necessary.
- Goal Achievement: The agent evaluates whether its goals have been achieved. If the goals are met, the agent may select new goals and repeat the process. If the goals are not met, it revises its plans and actions to continue working toward them.

Goal-Based Agents

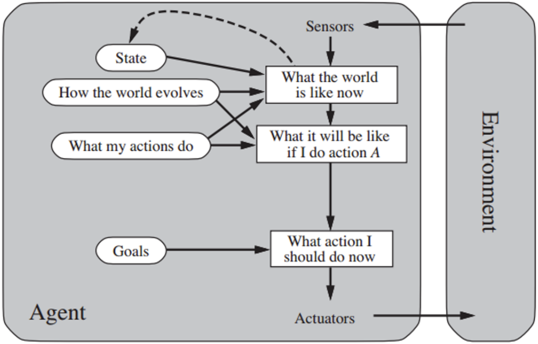
Goal-Based Agents are capable of more complex and flexible decision-making compared to Simple Reflex and Model-Based Reflex Agents.

They can adapt to changing circumstances, plan for the future, and pursue long-term objectives.

These agents are commonly used in various applications, including robotics, autonomous vehicles, game AI, and planning and scheduling systems.

Goal-Based Agents

Although the goal-based agent appears less efficient, it is more flexible because "the knowledge that supports its decisions is represented explicitly and can be modified."



Goal-Based Agents

Even Goal-Based Agents have limitations, especially in highly dynamic and uncertain environments. In such cases, more advanced agents like Utility-Based Agents or Reinforcement Learning Agents may be employed to handle complex decision-making scenarios.

Utility-Based Agents

A Utility-Based Agent operates by assessing the desirability or utility of different actions and selecting the action that maximizes its expected utility.
These agents are particularly useful when making decisions in situations with multiple conflicting objectives or when dealing with uncertainty.

Utility-Based Agents

Utility-Based Agents are similar to Goal-Based agents but provide extra feature of Utility Measurement.
Utility-Based Agent acts not only on goal but also the best way to achieve the goal.
It is recommended when we have multiple possible alternatives.

Utility-Based Agents

Just having goals isn't good enough because often we may have several actions which all satisfy our goal so we need some way of working out the most efficient one.

A utility function maps each state after each action to a real number representing how efficiently each action achieves the goal.

This is useful when we either have many actions all solving the same goal or when we have many goals that can be satisfied and we need to choose an action to perform.

Utility-Based Agents

Here's how a Utility-Based Agent typically works:

- Utility Function: The agent defines a utility function or utility model. This function assigns a numerical value (utility) to each possible outcome or state that can result from taking different actions. The utility represents the agent's preference or desirability for each outcome.
- Perception: The agent perceives the environment through sensors, collecting information about the current state and available actions.
- Action Evaluation: The agent evaluates the expected utility of each possible action by considering the current state, potential outcomes, and the utility function. It calculates the expected utility by taking into account the probabilities of different outcomes.

Utility-Based Agents

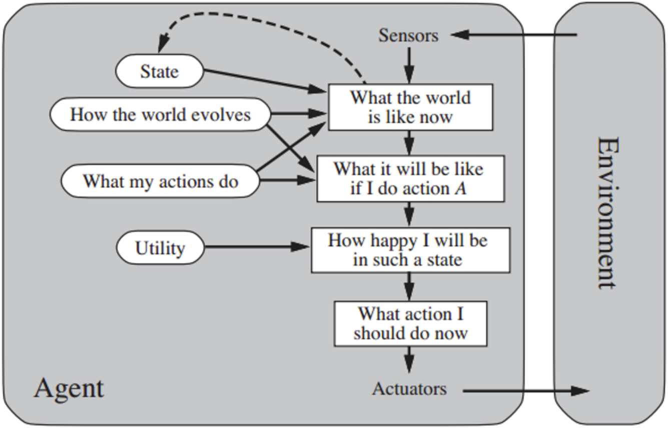
[Continued]

- Action Selection: The agent selects the action that maximizes its expected utility. In other words, it chooses the action that leads to the most desirable outcome based on its utility model.
- Action Execution: The agent executes the selected action and updates its perception of the environment.
- Feedback and Learning: Over time, the agent may learn and adapt its utility function based on feedback from the outcomes of its actions. This allows it to make more informed decisions in the future.

Utility-Based Agents

When there are multiple possible alternatives, then to decide which one is best, utility-based agents are used.
They choose actions based on a preference (utility) for each state.
Sometimes achieving the desired goal is not enough. For example we may look for a quicker, safer, cheaper trip to reach a destination.
Agent happiness should be taken into consideration.
Utility describes how "happy" the agent is. Because of the uncertainty in the world, a utility agent chooses the action that maximizes the expected utility.
A utility function maps a state onto a real number which describes the associated degree of happiness.

Utility-Based Agents



Utility-Based Agents

Utility-Based Agents are particularly valuable in decision-making scenarios where there are trade-offs between different objectives or where the environment is uncertain.
They are commonly used in fields such as economics, game theory, resource allocation, and artificial intelligence for tasks like path planning, resource scheduling, and multi-objective optimization.