CE213 Artificial Intelligence – Lecture 20

Intelligent Agents

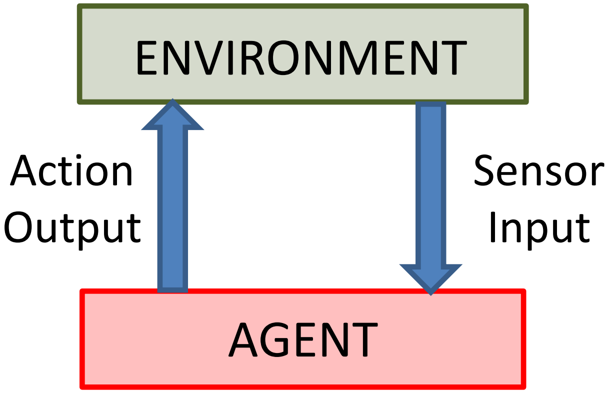
Integration of various AI approaches to
perception / environment understanding (NN),
state-action mapping (reinforcement learning, NN),
and decision making (decision tree, expert system, NN).

Architectures for Intelligent Agents and Multi-agent Systems

What Are Agents?

An agent is a system (hardware or software) that is

- situated in an environment
- capable of autonomous action in that environment to achieve goals or objectives



Autonomy (autonomous action) implies that it can act without the intervention of humans or other agents.

Agents must be able to:

- **perceive/understand their environments**
- **act upon their environments**

Basic functions of an agent are perception, decision making, and action.

Therefore, various AI methods may be required in an agent.

Simple Examples of Agents

Central heating thermostat

Environment:	Building whose heating it regulates.
Sensor:	Temperature sensor
Actions:	Turn heating on
	Turn heating off

Pressure cooker safety valve

Environment:	Pressure cooker and its surroundings.
Sensor:	Pressure sensor
Actions:	Open valve
	Close valve

Simple Examples of Agents (2)

New e-mail demon

Environment:	Operating systems and data structures
Sensor:	Unread message checker (software)
Actions:	Initiate audio warning signal

Simple spinal withdrawal reflex

Environment:	Animal’s body and surroundings
Sensor:	Pain sensors in skin
Actions:	Activate flexor muscles

Of course, robots are agents in general.

Intelligent Agents

What agents are Intelligent?

Criterion 1: Whether they can pass the Turing Test

This is too demanding to be a useful criterion

Criterion 2: Whether they are capable of ***flexible*** **autonomous behaviour**

What do we mean by "flexible"?

- They are Pro-active:

  Can take the initiative in order to achieve goals.

- They are conditionally reactive:

  Can react in a timely fashion to changes in environment by selecting appropriate response from a range of alternatives.

- They are socially interactive:

  Can interact with other agents (possibly including humans) to achieve collective goals.

Agents and Expert Systems

Are expert systems intelligent agents?

Most people would say "No" because:

They do not perceive their environments directly: they rely on the human user.

They do not act directly on the environment: they give advice to the user.

However, some expert system applications could be regarded as examples of intelligent agents.

e.g., expert systems equipped with sensors and actuators for real-time monitoring and control.

A Formal Representation of Agents

Environment (sensor input)

Let the environment be a set of states **S** where

$$\mathbf{S} = \{s_1, s_2, \ldots..\}$$

At a given instant, the environment is in one of these states.

Actions

Let the actions available to the agent be a set of actions **A** where

$$\mathbf{A} = \{a_1, a_2, \ldots..\}$$

Agent

Then the *agent* can be viewed as a function that maps a **sequence of environment states S*** to actions: (e.g., neural networks, control policies, etc.)

$$\mathbf{S^*} \rightarrow \mathbf{A}$$

Environmental Change (action output)

The effect of actions and other sources of environmental change can then be modelled as a function:

$$\mathbf{S} \times \mathbf{A} \rightarrow \mathbf{S}$$

Types of Intelligent Agents

Reactive Agents

Decision making is implemented as a direct mapping from ***current*** situation to action:

purely reactive agent : $\mathbf{S} \rightarrow \mathbf{A}$

(**S** rather than **S***)

A purely reactive agent has no memory of earlier environment states and thus responds only to the immediate situation.

(Similar to the Markov decision process described in reinforcement learning)

Types of Intelligent Agents (2)

Agents with Internal State

An agent that has some form of memory can base its action choices on both the current environment and its internal state.

We could represent this as:

$$agent : \mathbf{S} \times \mathbf{M} \rightarrow \mathbf{A}$$

where **M** is the agent's set of internal states (memory).

However, it is clearer what is happening if we represent the perception and action of the agent separately as follows:

$$agent\_perception : \mathbf{S} \times \mathbf{M} \rightarrow \mathbf{M}$$

$$agent\_action : \mathbf{M} \rightarrow \mathbf{A}$$

Subsumption Architecture

A reactive agent architecture developed by Rodney Brooks at MIT
(https://en.wikipedia.org/wiki/Subsumption_architecture)

Two main ideas:

1. ***Task Accomplishing Behaviours***

   Agent's decision making is realised through a set of behaviours that determine or select individual actions.

   Each of these behaviours is a simple response to the current situation.

2. ***Subsumption Hierarchy***

   Behaviours are arranged in *layers*.

   Lower layers can inhibit higher layers.

   Thus when two or more behaviours are appropriate for the current situation, the one in lowest layer will be chosen. (conflict resolution)

   Higher layers represent more abstract aspects of the problem domain or relate to longer term objectives.

Will have an example later.

Pros and Cons of Reactive Agents

Advantages

- *Cheap*: Computational power required for each agent is low.
- *Robust*: Loss of single agent does not seriously disrupt operation of entire set (no centralised communication/control).

Disadvantages

- All decisions are based on purely local information, so agents must be able to obtain sufficient information locally to select best action.
- All behaviours are "short-term". Cannot take account of information regarding states encountered earlier.
- Learning from experience is not possible (no memory).

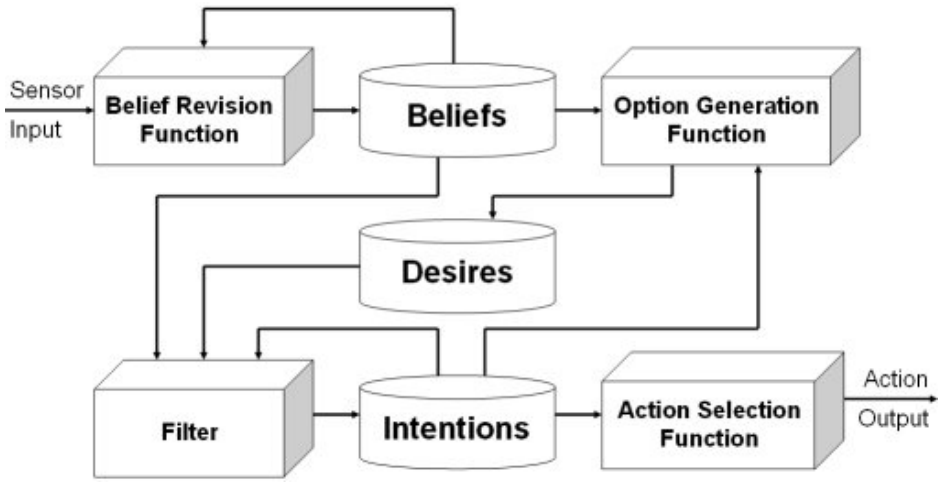
Architecture for Agents with Internal State

(An Example Only, for self study)

Belief-Desire-Intention (BDI) Architecture

Originally proposed by Bratman in 1987, trying to explain the human reasoning with the following attitudes: belief, desire and intention.

Decision making depends on the **manipulation of various internal data structures** representing beliefs, desires and intentions of the agent.



S x M → A or S x M → M, M → A

Belief: informational state of the agent (memory)

Desire: motivational state of the agent (memory)

Intention: deliberative state of the agent (memory)

They are updated continuously through perception

Multi-agent Systems

An important (and very interesting) class of problems are those involving more than one agent operating in the same environment.

Interaction

They may ***cooperate*** or they may ***compete***.

If they cooperate they may need to *construct plans* to achieve collective goals. Such planning may be done *centrally* or *distributed* across the agents.

If they compete they may need to *negotiate* to resolve conflicts.

Communication

In both cases, cooperate or compete, they are likely to need to *communicate* with each other.

Such communication can be either

Direct communication:

Agents exchange messages with each other.

or

Indirect communication:

Agents communicate by acting upon the environment.

e.g., making marks in the environment.

An Example: Planetary Exploration
(Multi-agent system in Subsumption Architecture)

Scenario:

- The system is designed based on foraging behaviour of ants.
- A set of autonomous vehicles (agents) are landed on Mars.
- Their task is to collect samples of a rare type of rock and bring them back to the mothership.
- Locations of the rock are not known in advance, but tend to fall in clusters.
- No map of the planet is available.
- The vehicles cannot communicate directly with each other.

Planetary Exploration (2)

Navigation

To aid vehicles in returning, the mothership broadcasts a steady signal.

The intensity of this signal reduces as the distance from the mothership increases.

Vehicles can thus find their way back by moving in the direction that most increases the signal strength.

This navigational mechanism is called a ***gradient field***.

Planetary Exploration (3)

Individual Behaviours (represented as production rules)

The behaviour in lowest layer should enable an agent to avoid obstacles:

IF	**detect an obstacle**	
THEN	**change direction**	**[*Layer 1*]**

The next two important behaviours ensure that an agent that has found rock samples will take them back to the mothership (base):

IF	**carrying samples**	
AND	**at base**	
THEN	**drop samples**	**[*Layer* 2]**

Planetary Exploration (4)

IF **carrying samples**
AND **not at base**
THEN **travel up gradient** **[*Layer 3*]**

(Note that 'obstacle avoidance' can inhibit 'return to base' behaviours.)

The next behaviour ensures that an agent picks up samples that it finds.

IF **detect a sample**
THEN **pick up sample** **[*Layer 4*]**

(*Note that by layering architecture this behaviour is inhibited once the agent has picked up samples*)

…… (some other behaviours, e.g. for communication)

The final behaviour in the highest layer, with lowest priority, produces exploratory behaviour.

IF **true**
THEN **make random move** **[*Layer 10*]**

Planetary Exploration (5)

Cooperation (no competition)

The set of behaviours designed so far enables the vehicles (agents) to operate independently to perform the sample collection task.

It lacks a means to enable them to cooperate:

That is, to pass on the information they have discovered about where samples are to be found.

We can address this limitation by providing them with a simple facility for **indirect communication**.

We therefore assume that the vehicles can also:

- Drop crumbs of rock samples
- Detect crumbs of rock samples
- Pick up crumbs of rock samples

Planetary Exploration (6)

First we can extend one of the existing rules, so that agents will lay a trail of crumbs as they return to base:

IF	**carrying samples**	
AND	**not at base**	
THEN	**drop 2 crumbs**	
	travel up gradient	**[*Layer 3*]**

Then we can add a new rule for dealing with the situation when crumbs are encountered during exploratory behaviour:

IF	**sense crumbs**	
THEN	**pick up 1 crumb**	
	travel down gradient	**[Layer 5]**

More behaviours/rules?

Discussions: What are the differences from general expert systems?

How about perception? – Detection of obstacles, samples, crumbs, …… (neural networks may be applied here)

Summary

What Are Agents

Simple examples of agents

Intelligent Agents

Pro-active, Conditionally reactive, Socially interactive

A Formal Representation of Agents

Reactive agents, Agents with internal state

Subsumption Architecture

Task-accomplishing behaviours arranged in layers

Pros and Cons of Reactive Agents

BDI Architecture for Agents with Internal State

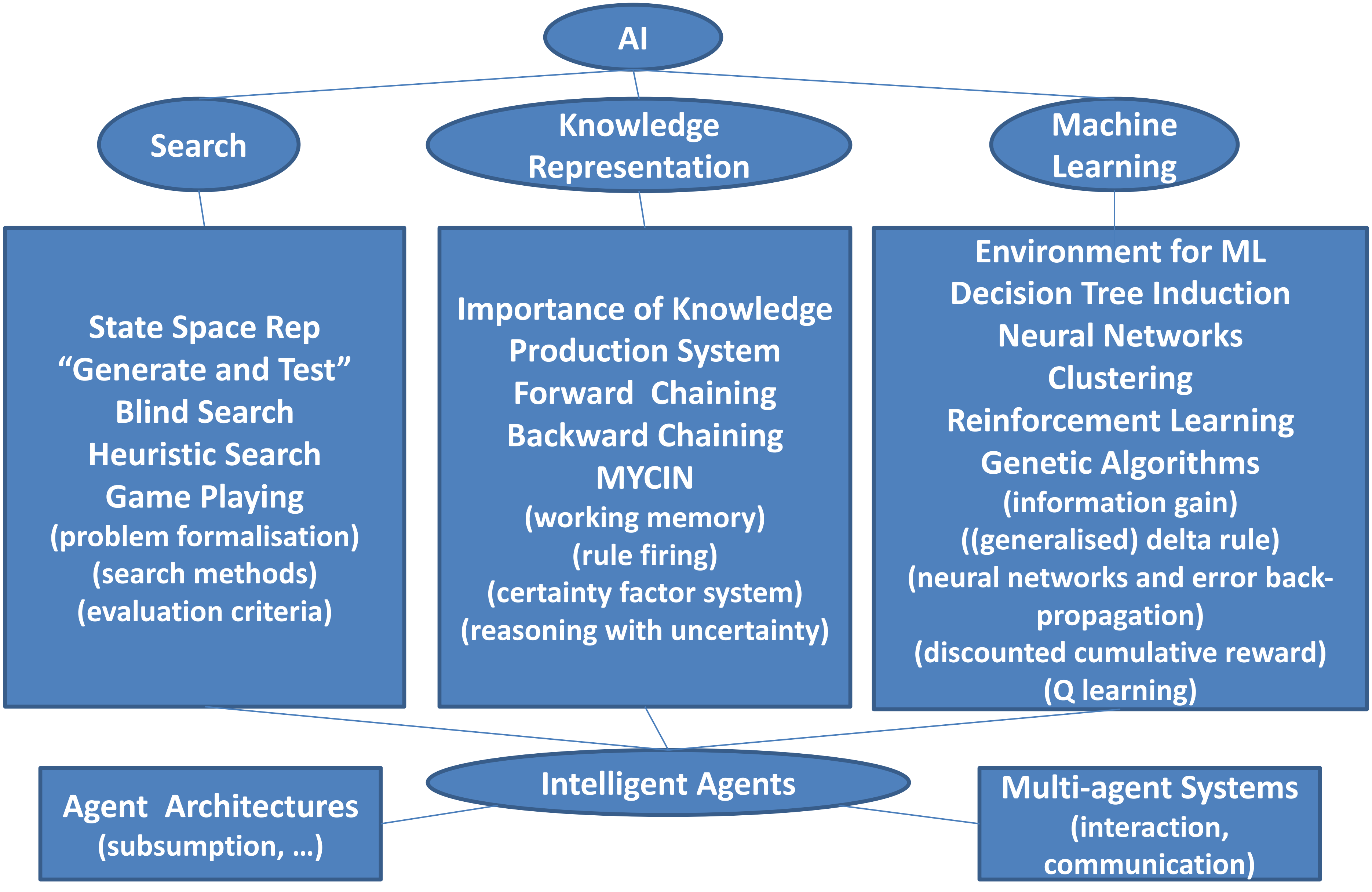
Multi-agent Systems

Interaction: Cooperation or competition

Communication: Direct or indirect

Planetary exploration example

Basic AI Concepts and Methods



If you want to learn more about AI ……

Hopefully, CE213 has laid a solid foundation for possible future modules:

CE215 Robotics

CE217 Computer Game Design

CE310 Evolutionary computation and Genetic Programming

CE316/CE866 Computer Vision

CE801 Intelligent Systems and Robotics

CE802 Machine Learning and Data Mining

CE811 Game Artificial Intelligence

CE889 Artificial Neural Networks and Deep Learning.