COMPSCI4004/COMPSCI5087 AI (H/M)

Week 2: Introduction and Foundations

Debasis Ganguly [1]

[1]University of Glasgow, Glasgow, UK

September 29, 2025

Overview

Lecturers and Time table

- **Lecturers:**
  - Dr. Debasis Ganguly, `Debasis.Ganguly@glasgow.ac.uk` (course coordinator).
  - Dr. Edmond S. L. Ho, `Shu-Lim.Ho@glasgow.ac.uk`
- **GTAs (for lab support):**
  - Shaul Ashkenazi
  - Kaiwen Zheng
- **Lectures**: Mondays: 15:00-17:00 at Rankine - Room 408.
- **Lab sessions**: Mondays: 17:00-18:00 and Tuesdays 17:00-18:00
  - you will be allocated a specific 1 hour timeslot, which should appear in your timetable.
- **Open Hours**:
  - Edmond Ho - Friday 12 noon - 1 PM, SAWB 402, Sir Alwyn Williams Building.
  - Debasis Ganguly - Friday 2 PM - 3 PM, M111, Sir Alwyn Williams Building.

Course Information

- AI (H and M): Overview of **Intelligent Agent Design**.
- Fundamental concepts of AI.
  - We'll explain various stages and complexities of an agent-driven model that **interacts with an environment** and makes a **sequence of rational decisions**.
- Non-examinable materials:
  - Recent advancements in AI.
  - Responsible AI (explainability, trustworthiness and fairness).
- Labs:
  - Labs in the week will be based on the lecture notes.
  - Labs **aren't graded** but you should complete the exercises. We will release the solutions the next day at 7 PM.

Exams and Coursework

- Exams (70%):
  - Exams will be closed book and invigilated.
  - A crib-sheet containing formulae and other essential stuff will accompany the exam paper.
- Coursework (30%):
  - A formative assessment which we will release after the week 3 lecture.
  - Gives you a scope to start early, build your initial simple solution based on what will be taught in Week 3.
  - You can then make your system better in an incremental manner by applying the concepts learned from the subsequent weeks.
  - Deliverable: code and report (**analysis more important**).

Intended outcomes

1. Demonstrate familiarity with the history of AI, and understand the **potential and limitations** of the subject in its current form.
2. Explain the basic components of an **intelligent agent**, and be able to map these onto other specific subjects such as information retrieval, computer vision, human-computer interaction etc.
3. Discuss basic issues in planning and **rational decision making**.
4. Explain and apply **search-based problem-solving** techniques.
5. Formulate and apply **Bayesian networks** in modelling and planning.
6. Explain and apply **utility theory** as a probabilistic framework for **rational decision making**.
7. Explain and apply **reinforcement learning** techniques to learn from rewards and observations.

Road Map of Weekly Teachings

2. Introduction and Foundations
3. Deterministic problems - search and optimisation
4. Stochastic Problems, Probability and Knowledge Representation
5. Decision-making under uncertainty
6. Sequential decision-making under uncertainty - MDPs
7. Learning from rewards and observations - basic Reinforcement Learning
8. Learning from rewards and observations - Reinforcement Learning with linear and non-linear function approximation
9. Learning from rewards and observations - improved DQN and policy search (with function approximation)
10. Generative AI applications (Planned Industry talk)
11. Practical Problem Solving and Revision

Four Different Viewpoints

Thinking Humanly	Thinking Rationally
Acting Humanly	Acting Rationally

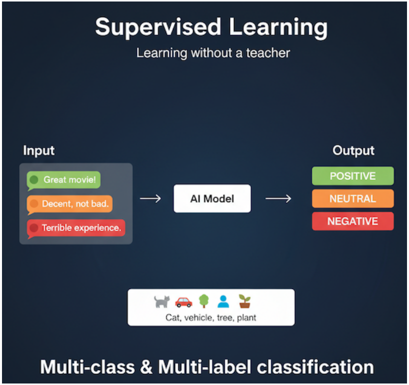
- Philosophical issues: whether machines can actually "think".
- Emphasis on **Acting** – a way to avoid philosophical debates.
- **Acting Humanly** - the Turing Test approach.
  - Humans don't always think rationally, e.g., humans have stereotype thoughts, humans gamble etc.
  - **Acting rationally** - is what we aim the AI to do!

Types of Learning in AI

Unsupervised learning (learning without a teacher)

- Example: Discover patterns within the data.
- Figure out which data is similar to what and different from others.

Types of Learning in AI



Supervised Learning

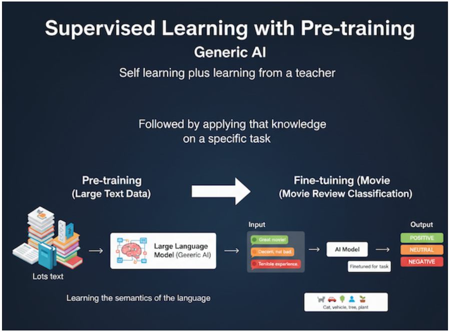
- Multi-class classification: Is this movie review positive, neutral, or negative?
- Multi-label classification: More than one class can be present in an instance, e.g., objects within an image.
- Update model parameter based on the examples.
- This course will cover the very basics. Covered at more depth in the ML course.

Types of Learning in AI

Reinforcement Learning

- Learning to be **adaptive** within an environment.
- Example: You want to find out your way out of a maze.

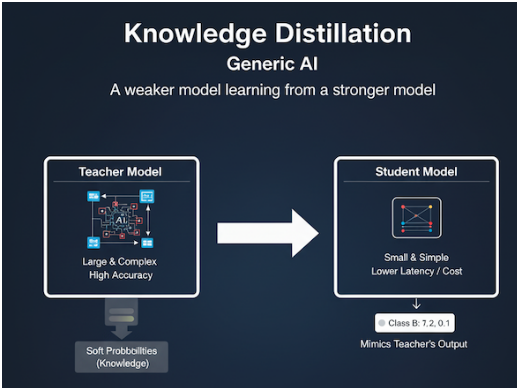
Types of Learning in AI



Artificial General Intelligence (AGI)

- Supervised Learning with Pre-training or Generic AI (self learning plus learning from a teacher)
- Learning the semantics of a language from lots of text (self/unsupervised learning)
- **Applying that knowledge on a specific task**, e.g., movie review classification.

Types of Learning in AI



Knowledge distillation

- Teach a weaker model from the predictions of a stronger model.

Example AI Systems

- **IBM's Deep Blue** defeats Garry Kasparov, the world champion in chess in 1997.
- Modern chess engines like Alpha-Zero, StockFish etc. are **much better than human players**: ≈ 3500 ELO as compared to ≈ 2830 of Magnus Carlsen!

Example AI Systems

- **IBM's Watson** system competed on Jeopardy! winning the first-place prize of 1 million USD.
- Leverages NLP/Information Retrieval and Knowledge-bases for effective Question Answering.

Example AI Systems

Can you point me to papers which discuss the challenges of test-case driven development?
Output the bibtex of the relevant papers.

Searched 4 sites

Several papers discuss the challenges of Test-Driven Development (TDD) in various contexts, focusing on issues like productivity, code quality, and the developer's learning curve.

1. **"Causal Factors, Benefits and Challenges of Test-Driven Development: Practitioner Perceptions"** by Jim Buchan et al. This paper explores the experiences of software teams using TDD, identifying challenges such as initial time overhead, difficulties in maintaining test suites, and the learning curve for developers unfamiliar with TDD. It suggests that while TDD can improve code quality, the upfront investment in learning and writing tests can slow initial productivity(ar5iv)(ar5iv).

- Open AI's Chat-GPT is a large *language model* (LLM) that is able to converse with humans or other LLMs.
- Leverages:
  - *Pre-training* on large volumes of text data
  - Represents context of words (tokens) as high dimensional vectors.
  - A step towards Artificial General Intelligence.

Why is AI difficult

- AI has been successfully applied for specific tasks, achieving super or comparable human performance.
  - Examples: Game playing (Chess/ Atari games), question answering (Chat-GPT), self-driving cars etc.
- But are the machines really "intelligent"?
- What is the definition of "intelligence"?

> "A very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience. It is not merely book learning, a narrow academic skill, or test-taking smarts. Rather, it reflects a broader and deeper capability for comprehending our surroundings..."

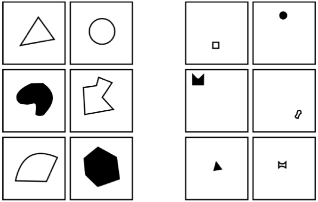
- We have progressed well on the aspects colored as blue.
- What about other more general tasks, such as the ones shown in red?

Bongard Problems

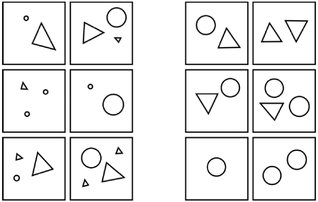
- Invented by the Russian computer scientist Mikhail Moiseevich Bongard.
- Popularized by Douglas Hofstadter in his Pulitzer prize winner - *Godel, Escher and Bach*.

Task

- *Explain* (in language), *why* the images on the left different from those in the right.
- Tests the abstract thinking capacity.



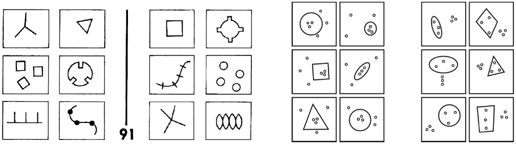
Large figures vs. Small figures



Small figure present vs. No small figure present

Characteristics of (Human) Intelligence

- Different levels of abstraction.
  - What combinations of attributes to use to define an object.
  - Some are more fine-grained (e.g., number of corners, lines etc.) than others (e.g., convexity).
- Moving back and forth between these representations to define how are objects similar and how are they dissimilar, specific to a task.



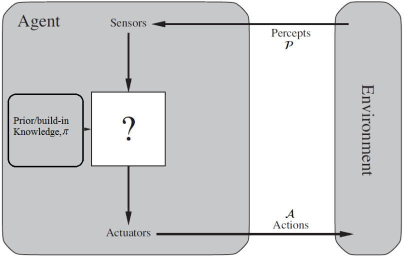
  - **Left:** BP denotes an *abstract property* for the understanding of numbers 3 and 4. More fine-grained concepts of corners, lines, wedges don't work. **Right:** An abstract concept of density is required.

What is possible today (2024)?

- Drive safely along a curving mountain road?
- Drive safely along University Avenue in the first week of the semester?
- Buy a week's worth of groceries on the web?
- Play a decent game of - Bridge/Go/Chess?
- Discover and prove a new mathematical theorem?
- Write an intentionally funny story?
- Give competent legal advice in a specialized area of law?
- Converse successfully with another person for an hour?
- Perform a complex surgical operation?
- Unload any dishwasher and put everything away?
- Play a decent game of table tennis?
- Explain your feeling and emotions to others?
- Learn, adapt and develop over several decades?
- Learn a new motor skill from a few examples?

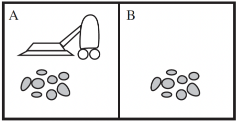
Key question: How should we study, design and build intelligent agents that behave rationally?

- What is an agent?
  - An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators (includes humans, robots, chatbots, thermostats).



PEAS model

- It is useful to 'think' about any AI task as a **PEAS** model.
  - Performance measure: - Defines what "good behaviour" is in a specific context.
  - Environment: A specification of the physical (or virtual) environment the agent is expected to operate in.
  - Actuators: The types and physical properties of the actuators available to the agent. Limits what the agent can do.
  - Sensors: The types and physical properties of the sensors available to the agent. Limits what the agent can know about the environment.



Agent function

The agent function (as implemented by the agent program) maps from prior/built-in knowledge, π, and percepts, $\mathcal{P}$, to actions $\mathcal{A}$, i.e.,

$$f : \mathcal{P}, \pi \rightarrow \mathcal{A}$$

- **Percepts**, $\mathcal{P}$: Perceptual inputs, percepts, and sequence/history as reported by the sensors.
- **Actuators and actions**, $\mathcal{A}$: What ever means the agent has to influence the environment via its actuators (visual, physical, audio, computer commands, etc.)

Agent function (Contd.)

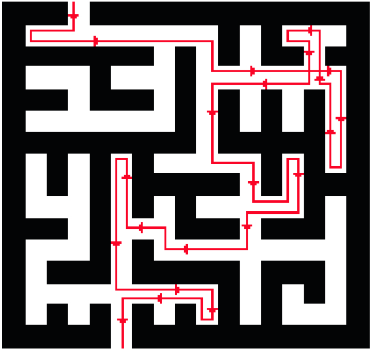
- **Prior knowledge**, π: Any hard coded constraints or knowledge about the environment (e.g. if temperature $<$ -40 degrees is not good).
- **Function**, f:
  - An abstract, external characterization of the agent by a mathematical function which can be represented by mathematical object such as a look-up table, a continuous or discrete function.
  - Implemented as a **agent program** and runs on a physical device.

Let's take up a quiz on Menti.com

Go to `mentimeter.com`; **use code '6131 7687'**.

PEAS view of existing AI models

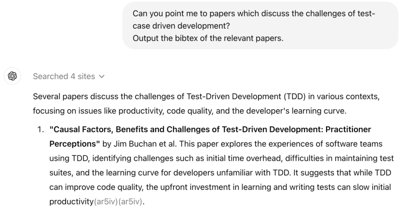
Maze finding



- Performance measure: **Minimize** #steps taken, or time spent in the maze.
- Environment: Size of the maze, start, goal, paths and obstacles.
- Actuators: Movement through a grid - virtually or physically.
- Sensors: Reacting to obstacle - virtually or physically.

PEAS view of existing AI models

Chat-GPT


Can you point me to papers which discuss the challenges of test-case driven development?
Output the bibtex of the relevant papers.

Searched 4 sites

Several papers discuss the challenges of Test-Driven Development (TDD) in various contexts, focusing on issues like productivity, code quality, and the developer's learning curve.

1. **"Causal Factors, Benefits and Challenges of Test-Driven Development: Practitioner Perceptions"** by Jim Buchan et al. This paper explores the experiences of software teams using TDD, identifying challenges such as initial time overhead, difficulties in maintaining test suites, and the learning curve for developers unfamiliar with TDD. It suggests that while TDD can improve code quality, the upfront investment in learning and writing tests can slow initial productivity(ar5iv)(ar5iv).

- Performance measure: **Maximize** - correctness, relevance, or **Minimize** - reading effort, misninformation of the answer.
- Environment: The virtual space of all possible answers (quantized in terms of tokens).
- Actuators: Generating a token conditioned on the input and what's generated before.
- Sensors: The API interface that gets a user's text.

PEAS view of existing AI models

Self-Driving Cars

- Performance measure: **Maximize** - safety, or **Minimize** - time to reach destination (performance measures can be conflicting with each other!).
- Environment: The surface on which the car is driven, the obstacles, road corners etc.
- Actuators: Brake, accelerator, gear.
- Sensors: Sequences of images captured, or other physical sensors like wetness of the road etc.

An agent should behave rationally

- What is rational behaviour? What does it mean to do a thing 'the right way'?
  - Objective answer: Consider the consequences (the 'P' of the PEAS model) of an agent's behaviour.
- For each possible percept (sequence), $\mathcal{P}$, a **rational agent** selects an action (sequence) which is expected to maximize its performance measure given the evidence provided by the percept (sequence) so far and whatever prior/built-in knowledge the agent has.

Rationality in the vacuum world

EAS of the vacuum world AI task

- **E**: two rooms (no prior knowledge about the prior likelihood of the dirt distribution).
- **A**: Left, Right, Suck
- **S**: Correctly identify if a room is clean.

Which 'P' leads to rationality?

1. +1 for sucking up a portion of dirt.
2. +1 for each clear square observed.
3. +1 for each clear square; -0.1 for taking an action due to battery usage.

Rationality in the vacuum world

Rule of Thumb

Choose a performance measure (P) on the basis of:

- Objective view-point: What is required in the environment.
- Subjective view-point: NOT on how an agent should behave.

- Which agents are rational?
  1. Cleans a square if dirty, otherwise moves to the other square over a period of 1000 time steps (say checks every 10 min).
  2. Cleans floors moving back and forth continuously for an hour and then goes to sleep for the day waking up after 23 hours.

What IS Rationality and what it IS NOT?

Rationality IS NOT omniscient

- An agent can't know the precise outcome of the action in the environment.
- It can only estimate the outcome based on previous percepts.

Rationality DOES NOT imply success

- Being rational does not imply success in solving the task.
- Example: Think about uncertain environments.

Rationality CAN LEAD to exploration, learning and autonomy

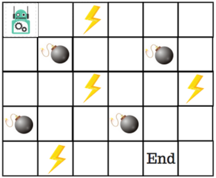
- Example: An irrational maze-finder can just continue to move back and forth; but then it doesn't learn the possible ways out of the maze.

Fully observable vs. partially observable

- **Fully observable**: Access to all relevant information via the sensors.
- **Partially observable**: If an agent acts based on noisy or broken sensors - or sensors simply do not capture the relevant information.

Deterministic vs. Stochastic

- **Deterministic**: The next state of the environment is fully captured by current state and the action to be carried out.
- **Stochastic**: Can not determine the next state based on current state and action due to randomness (or unknowns) in the environment.



Static vs. Dynamic

- Static: The environment never changes.
- Dynamic: The environment changes while we decide what to do, time matters!
  - Example: Ice melts (at a certain rate) on the frozen lake environment.
  - Example: Dirt accumulates with certain probability in the vacuum world environment.
- Semi-Static: The world is the same, but the performance score changes.
  - Example: Performance measure changed to maximizing battery life as opposed to just the room cleanliness in the vacuum world.

More environment types

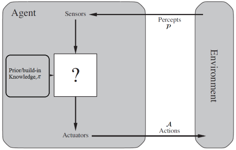
Discrete vs. Continuous

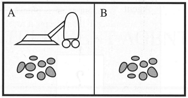
- Discrete: State of the environment are determined among a set of discrete possibilities (e.g. chess), discrete actions (e.g. move left or right), discrete percepts (e.g. dirty, not dirty)
- Continuous: The world has infinitely many states (e.g. temperature), actions are continuous, percepts are continuous (human vision).

Episodic vs. Sequential

- Episodic: Single actions based on current percept only, e.g., vacuum world.
- Sequential: Current action influences all future decisions, e.g., chess, maze finder.

Tabular (Rule-based) Agents



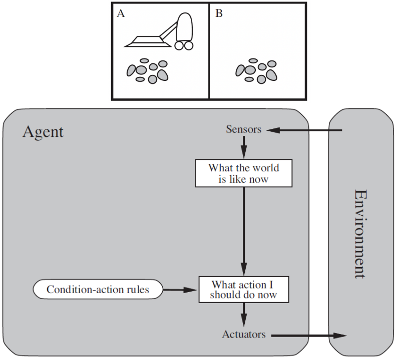


$\mathcal{P}$ Percept sequence	$\mathcal{A}$ Action
[*A*, *Clean*]	*Right*
[*A*, *Dirty*]	*Suck*
[*B*, *Clean*]	*Left*
[*B*, *Dirty*]	*Suck*
[*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮

- A pre-configured look-up table of state transitions.
- Keeps the **entire percept sequence** in memory.
- Feasible to define for a small-scaled task such as vacuuming two rooms.

$$[\text{A, clean}] \mapsto \text{Right}$$

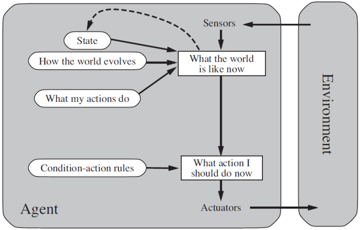
Reflex-based Agents



- If status=='Dirty' then 'Suck'
- If location=='A' then return 'Right'
- If location=='B' then return 'Left'

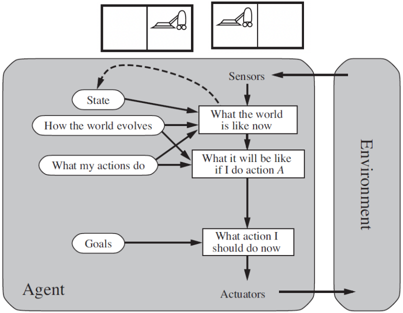
- The action is NOT a function of historical percepts but depends ONLY on the current percept (state).
- What happens if we only have a 'dirt' sensor, and no 'location' sensor?
  - What do we do if the state is clean? If we don't move we get stuck.
  - If we move then how can we figure out the direction? Moving left from 'A' will cause an infinite loop!
  - Does **randomizing actions** solve this?

Model-based Agents



- Reflex-based agent: Doesn't keep track of how the environment changes with an action;
  - e.g., Sucking dirt may introduce a new state where a room is neither completely clean, nor completely dirty.
- A **model-based agent** learns a mapping between the actions and consequences.

Goal-based Agents



- For some problems the goal state(s) is(are) known.
  - There are 2 goal states for the vacuum world - two clean rooms with the agent in any one of them.
- A rational agent should then perform actions that results in states that are '**closer**' to the goal state.
- We need then an evaluation function over states to measure this **closeness**.
  - For the vacuum world, how do we compute how far are we from the goal?

Limitations of Goal-based Agents

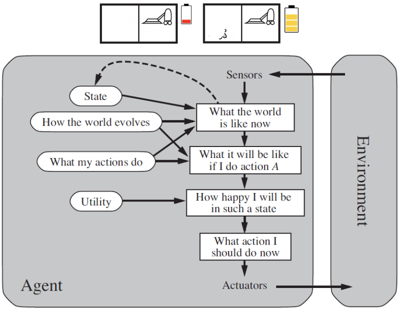
- What should an agent do when there are:
  - Multiple goals?
  - Conflicting goals?
  - Ill-specified goals (e.g., ‘user satisfaction’ for a conversational agent)?

Limitations of Goal-based Agents

- What should an agent do when there are:
  - Multiple goals?
  - Conflicting goals?
  - Ill-specified goals (e.g., ‘user satisfaction’ for a conversational agent)?

- Move towards the goal that’s ‘closer’ from the current state.
- Reach a trade-off (**Utility-based agents**).
- Try to maximize the performance as much as possible.

Utility-based Agents

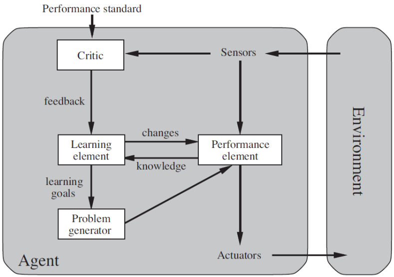


- A little bit of dirt in the other room with enough battery remaining to be recharged is a better goal than with a fully dead battery and two clean rooms.
  - Need a trade-off: which is what we do by defining a **utility function**.
- A rational agent should then perform actions that results in states that **maximize utility**.
  - For the vacuum world, what is a good utility function?
  - $u(state) = 0.9 \times cleanliness + 0.1 \times charge$?

A Generalized Learning Agent (Motivation)


- Can the most capable agents that we have looked thus far, viz. goal-based and utility-based agents that has been designed to perform well in the first environment also do well in the second?
- Why or why not (consider the following)?
  - Environments are stochastic.
  - State distribution is different.
  - What needs to be changed?

A Generalized Learning Agent (Design)



- **Performance Element**: Selects actions - similar to a static agent as we have seen so far.
- **Learning Element**: Finds improvements.
- **Critic Element**: Feedback from the environment which affects 'Learning Element'.
- **Problem Generator**: Choose sub-optimal paths to explore more about the environment to discover better actions in the long run.

A Generalized Learning Agent (back to the example)

- **Performance Element**: Utility function that minimizes the risk of falling down into a hole and maximizes the chance of getting the reward.
- **Learning Element**: Discover that two adjacent holes are more risky than a single one (this was not hard-coded into the utility function).
- **Critic Element**: Gives a high negative reward when an agent actually falls into a hole.
- **Problem Generator**: The agent needs to fall into holes (some risk taking ability) to improve its learning of maneuvering techniques around holes.

Summary

Now you know about:

- PEAS - Performance, Environment, Actuators, Sensors.
- Agent types - Journey from Reflex-based agents to Utility-based agents.
- Learning Agents - the most capable one.

To Do:

- Go through the lecture notes and try out the exercises from Chapter 2 of the AIMA book.
- Attempt the weekly quiz.
- Attend this week lab and carry out the exercises.

Anonymous Feedback for Continuous Monitoring

Join at menti.com | use code 1820 1009

Mentimeter

What is going right in the AI course or what is going wrong?

All responses to your question will be shown here

Each response can be up to 200 characters long

Turn on voting to let participants vote for their favorites