hogeschool vives

AI & Industry 4.0

Dr. Stefaan Haspeslagh - Vives University of Applied Sciences

25th – 29th | 2023 September | Sitia
https://1stathenatf.hmu.gr

1st ATHENA TECHNOLOGY FORUM

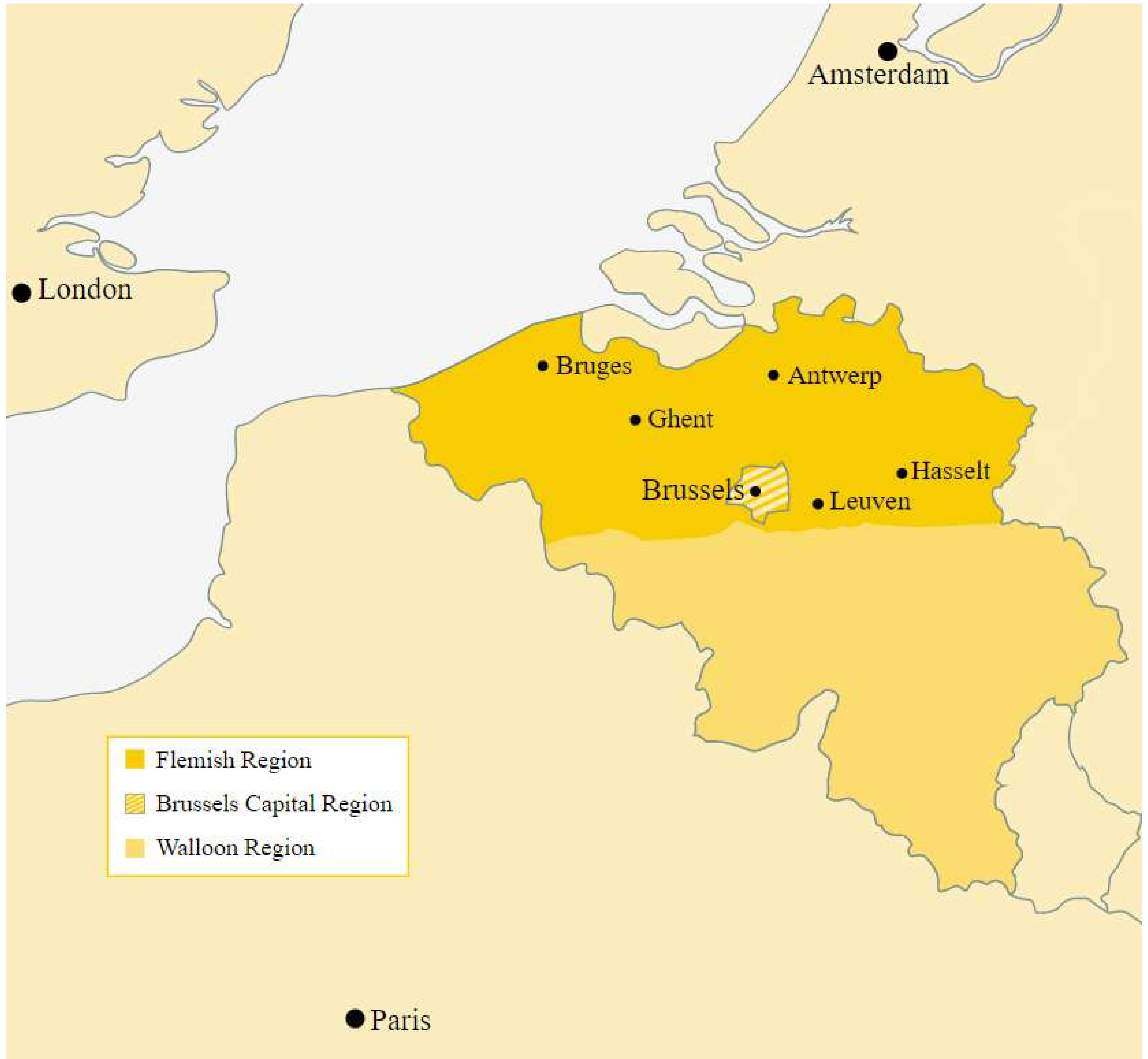
ATHENA EUROPEAN UNIVERSITY

Outline

- Vives?
- Industry 4.0? (For me)
- AI? (For me)
- AI & Industry 4.0 & Optimisation – Challenges we tackle @ Vives

Vives University of Applied Sciences

hogeschool vives

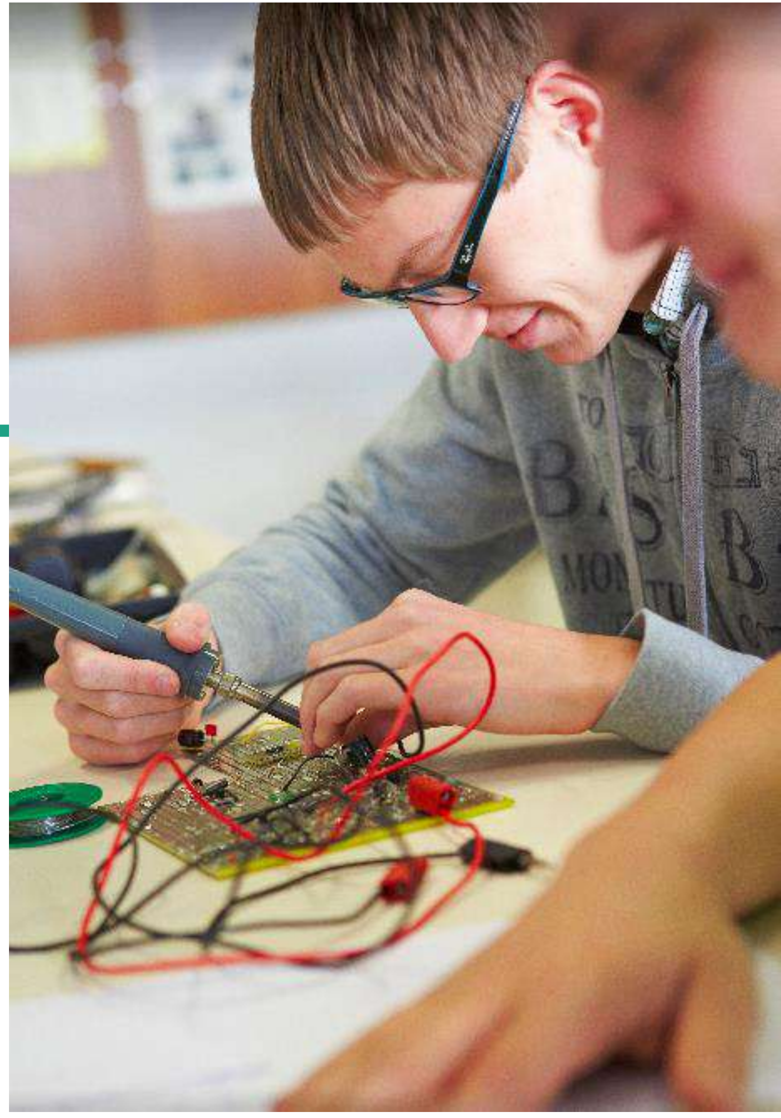


Belgium, where great minds grow

STUDY IN FLANDERS

www.studyinflanders.be

Vives in short

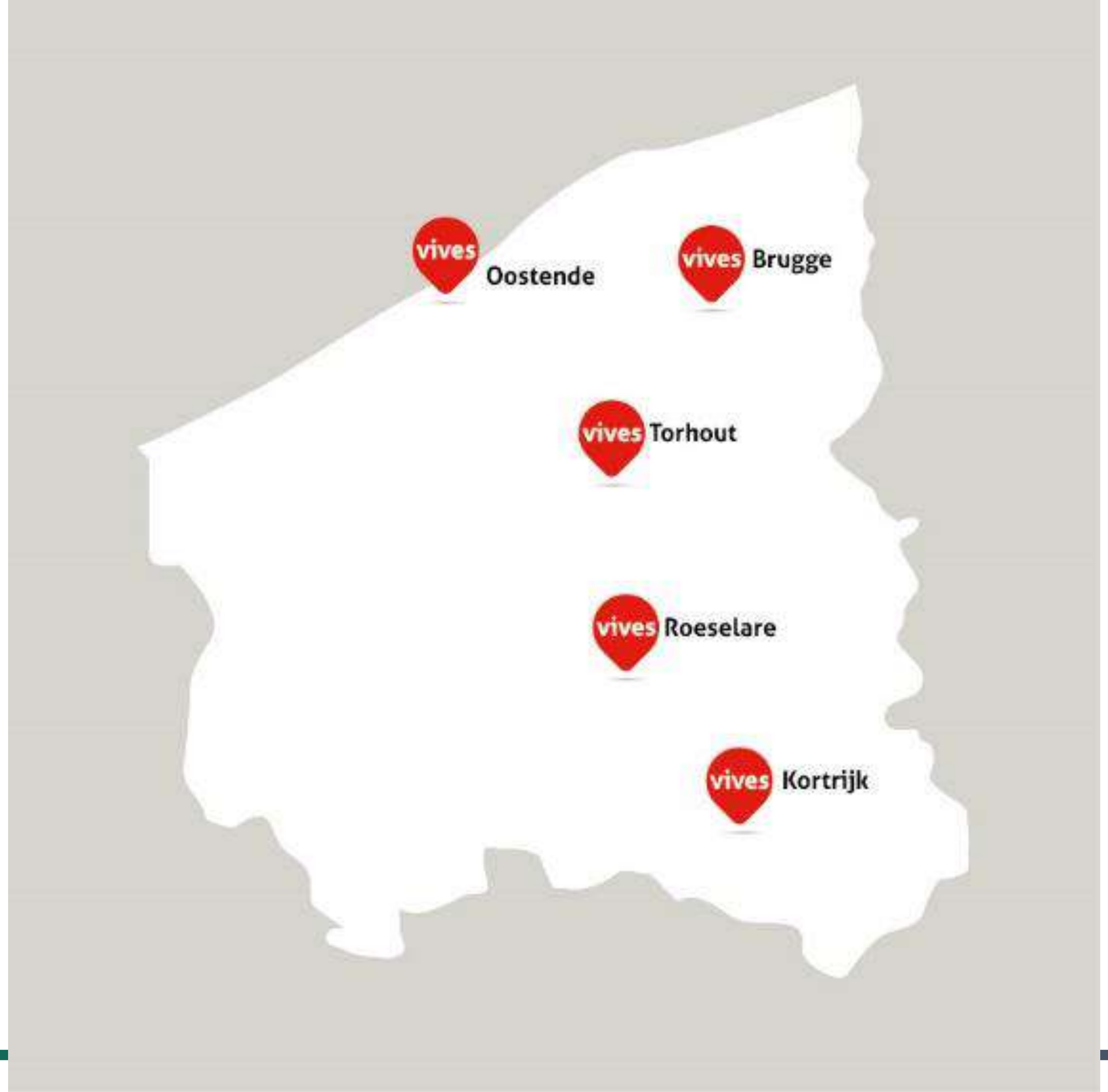


- The place to be for a broad choice and interesting offer!
  - 17.000 students
  - 900 lecturers
  - 163 researchers
  - 7 campuses in 5 cities
  - 4 faculties
  - 30 Bachelor's Degree programmes
  - 63 specialisations
  - 20 Associate Degree programs
  - >100 applied research projects
  - 6 research centres

Vives University of Applied Sciences

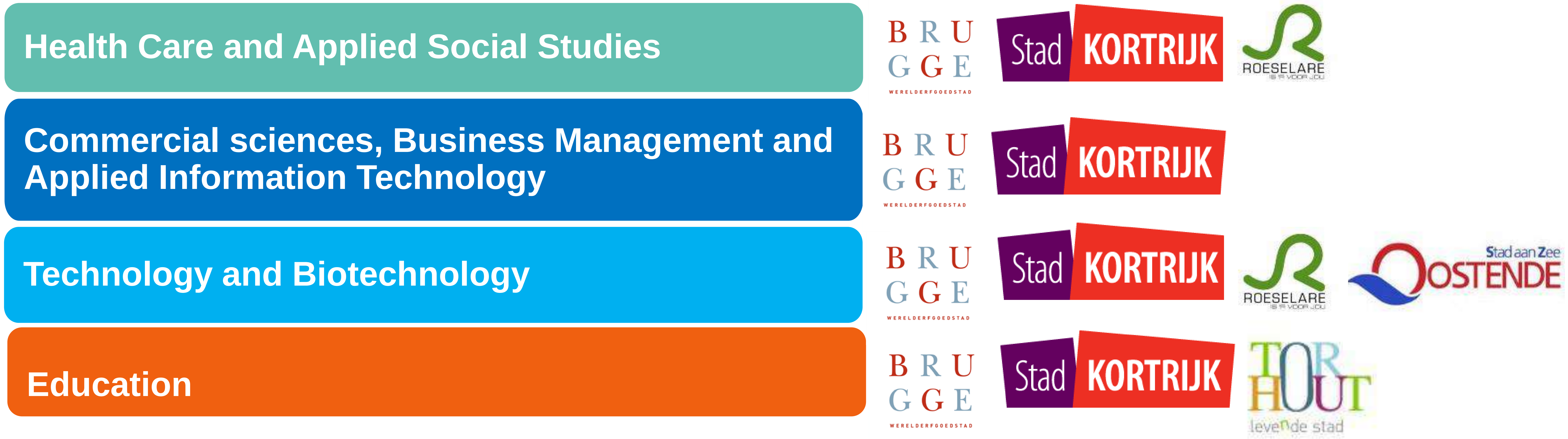
hogeschool vives

- 7 campusses in 5 cities



Vives University of Applied Sciences

- 4 faculties in 5 cities



Health Care and Applied Social Studies

Commercial sciences, Business Management and Applied Information Technology

Technology and Biotechnology

Education

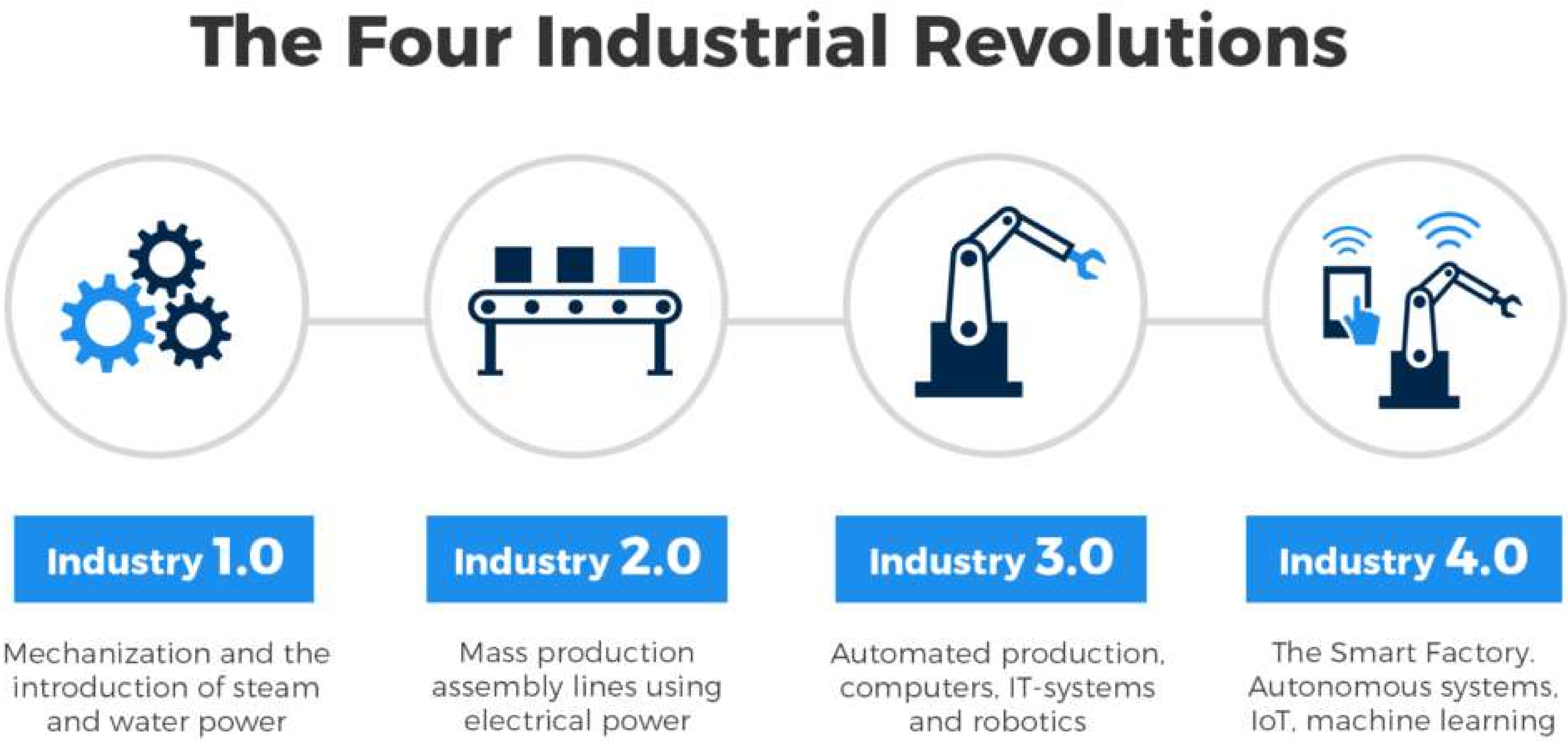
Who am I?

- Research manager @ Vives University of applied sciences and KU Leuven
  - Topics: AI – combinatorial optimisation - machine learning – deep learning
  - Applications in industry, biomedical sciences, health, transport, logistics, workforce planning, …

- Background in metaheuristic, exact and hybrid algorithms to tackle combinatorial optimisation problems (planning, scheduling, cutting, packing, vehicle routing, group composition, …)

- Lecturer in AI: optimisation – ML – DL

Industry 4.0



The Four Industrial Revolutions

Industry 1.0

Mechanization and the introduction of steam and water power

Industry 2.0

Mass production assembly lines using electrical power

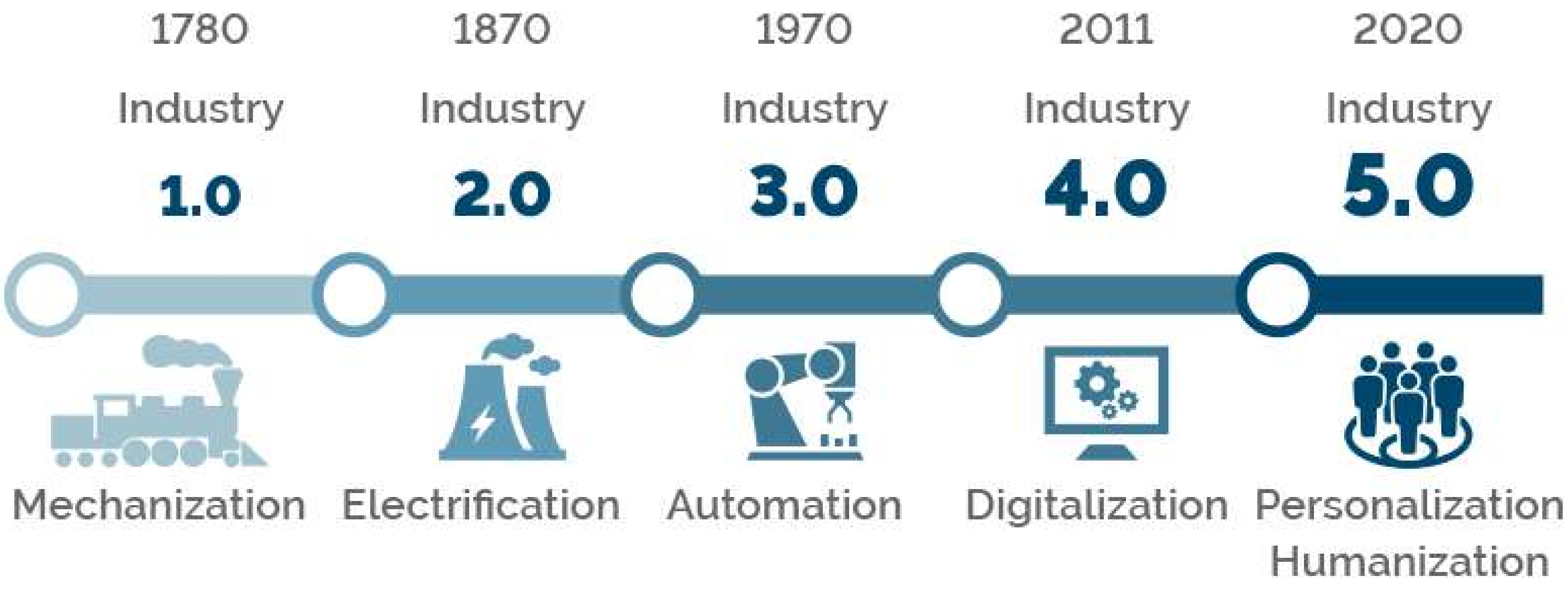
Industry 3.0

Automated production, computers, IT-systems and robotics

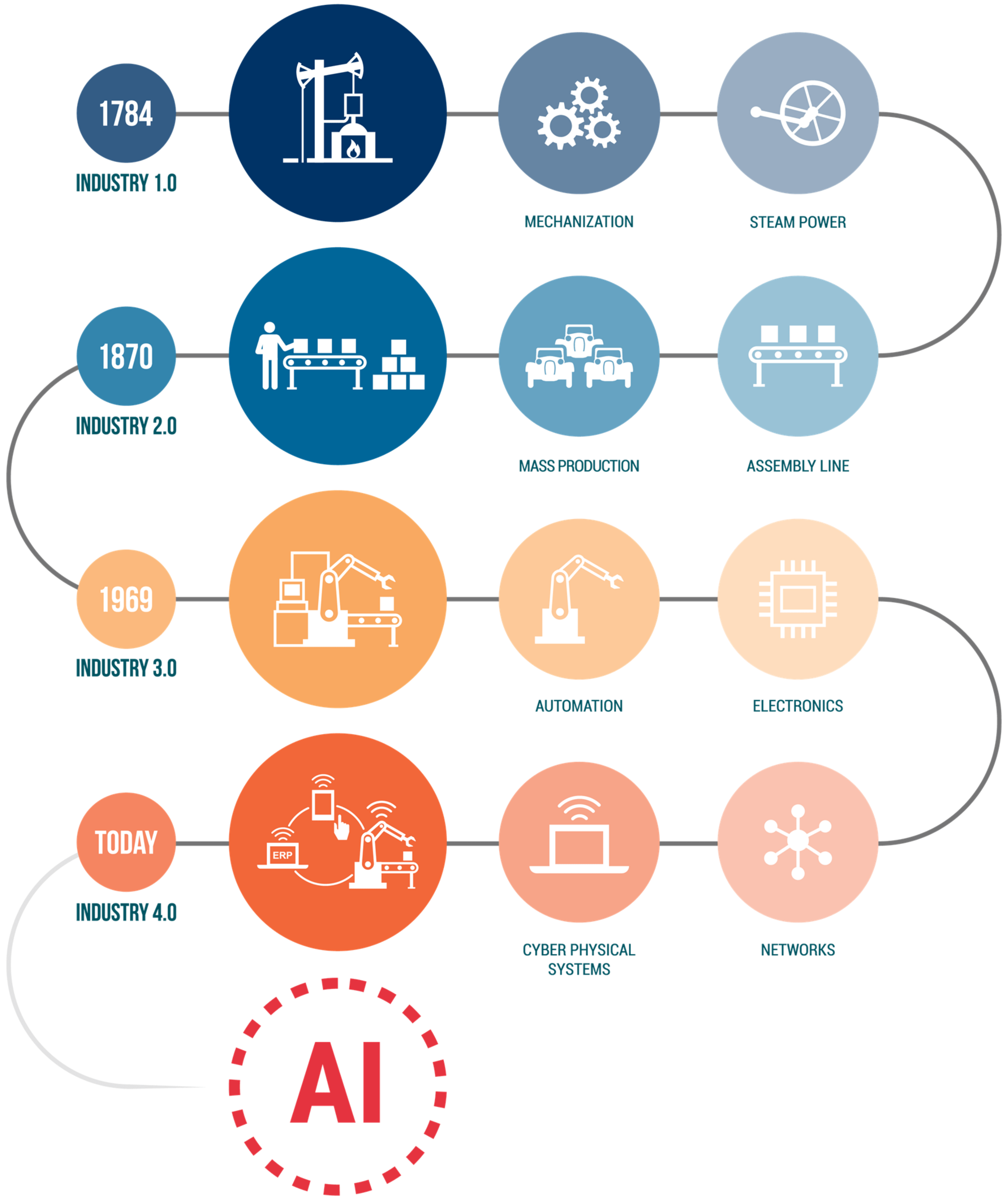
Industry 4.0

The Smart Factory. Autonomous systems, IoT, machine learning

Industry 5.0


1780
Industry
1.0
Mechanization
1870
Industry
2.0
Electrification
1970
Industry
3.0
Automation
2011
Industry
4.0
Digitalization
2020
Industry
5.0
Personalization
Humanization

Industry 4.0 → Industry 5.0



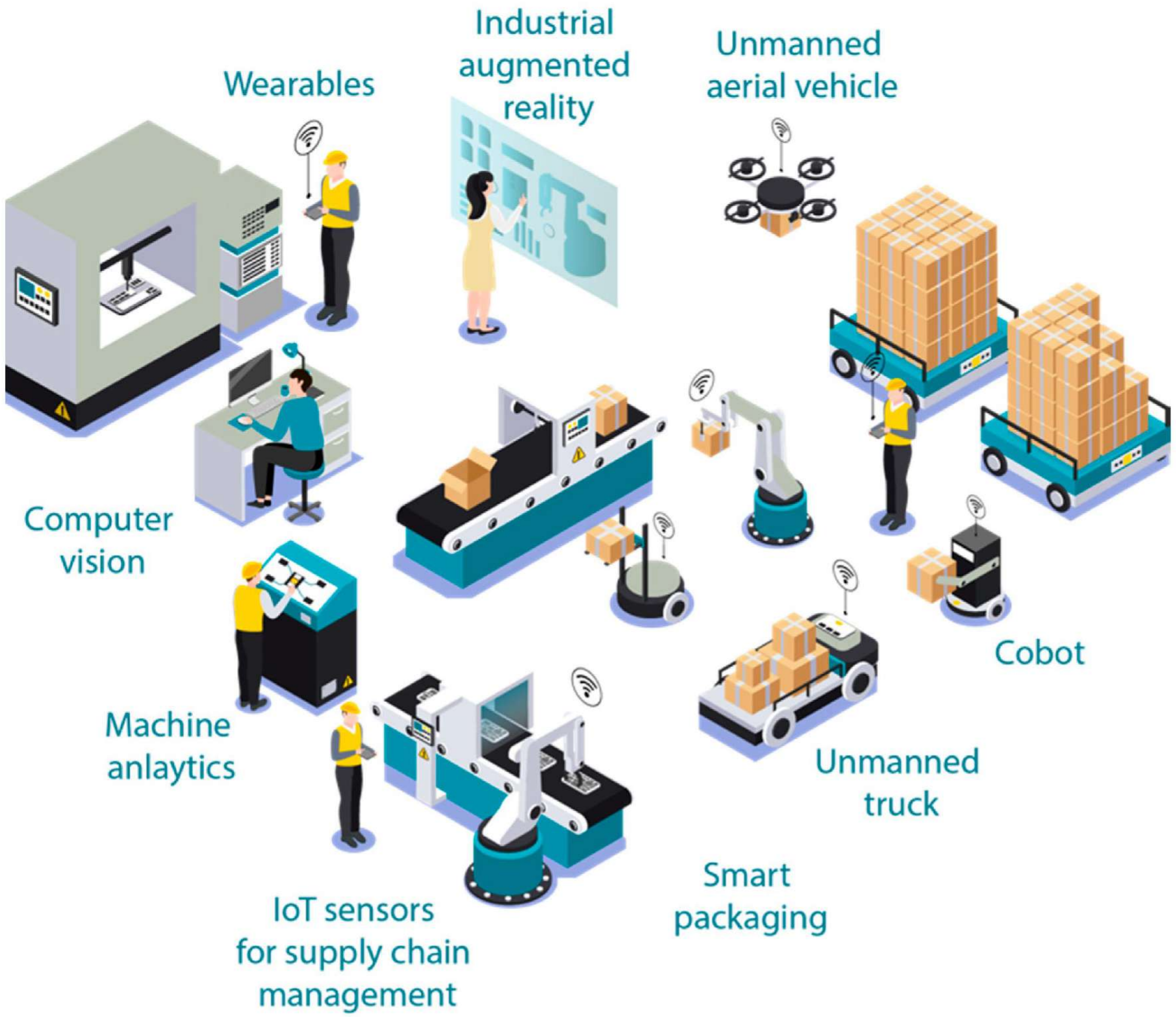
Industry 4.0 → 5.0: smart factory

- Customer-driven, highly customised manufacturing

- Need for:
  - Higher agility
  - Higher productivity
  - Higher sustainability

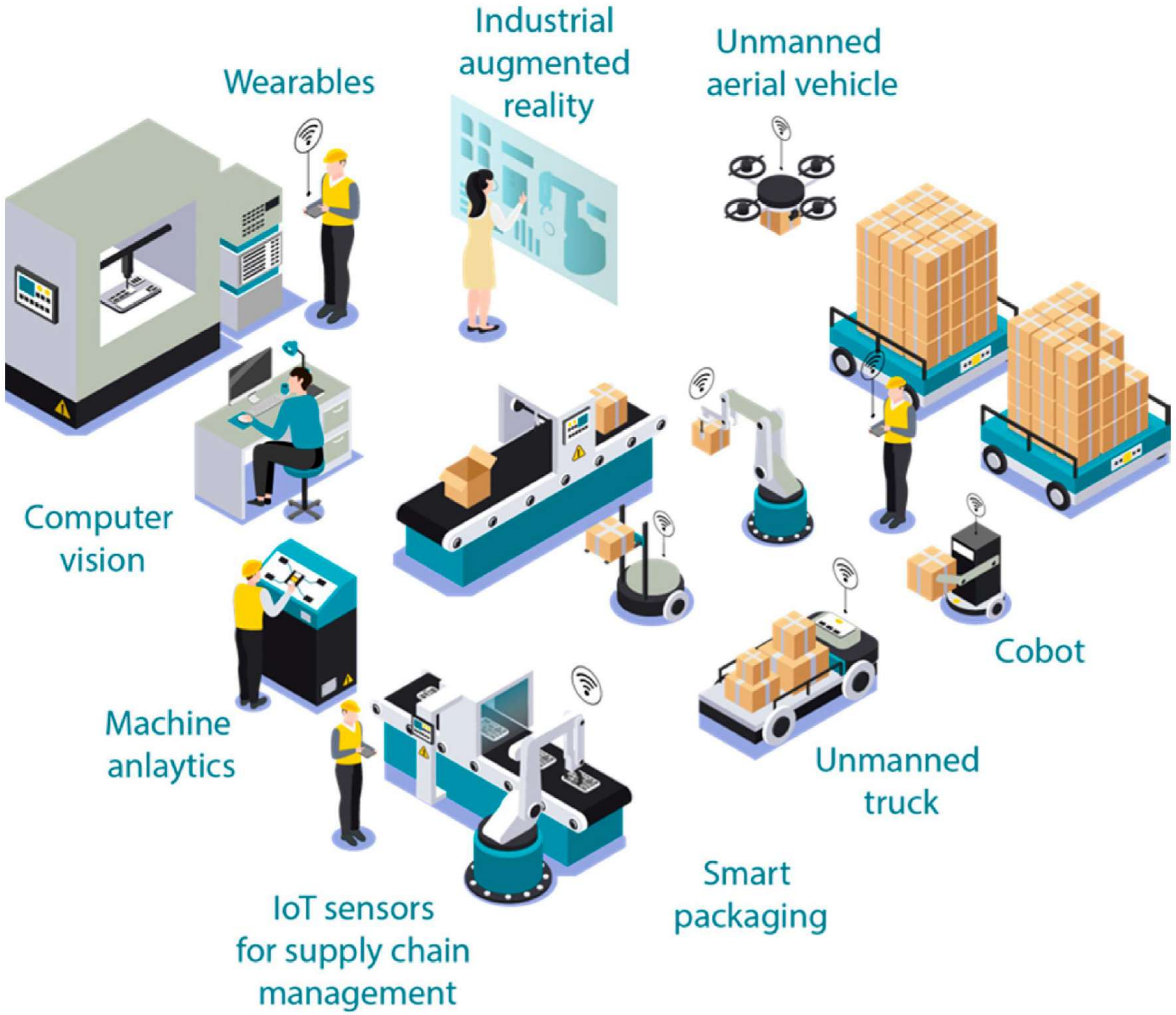
- Smart manufacturing: need for advanced *intelligent* systems to enable a dynamic response to *variable product demand*, along with a *real-time optimisation* along the *entire value chain*

- *Increased complexity! Need for further automisation!*

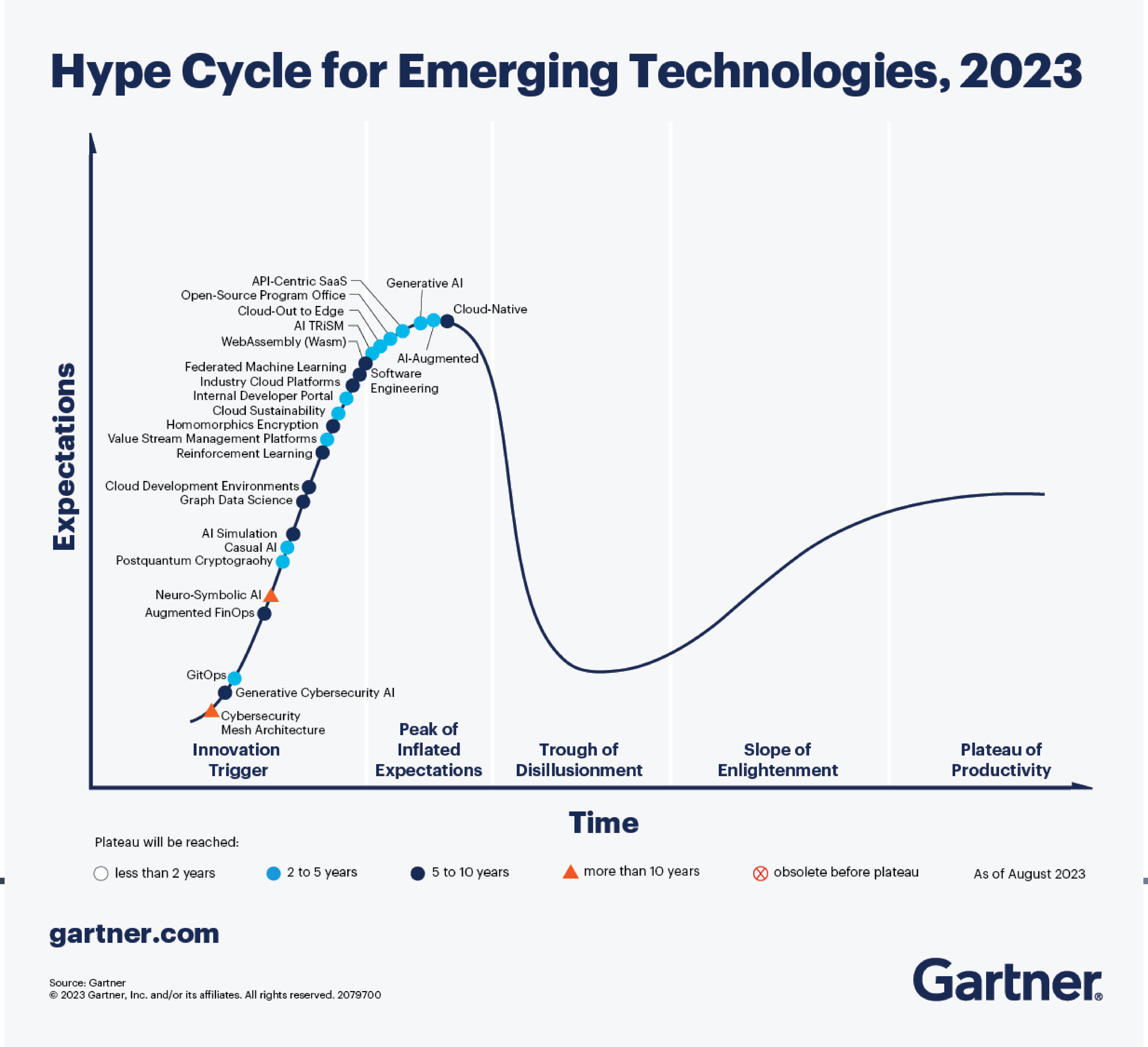


Industry 4.0 → 5.0: smart factory

- Some challenges:
  - Optimisation of production planning
  - Optimisation of workforce planning
  - Optimal routing of “unmanned trucks” and “cobots”
  - Forecast of demand
  - Synchromodal transportation of parts/products
  - …



AI – Gartner hype

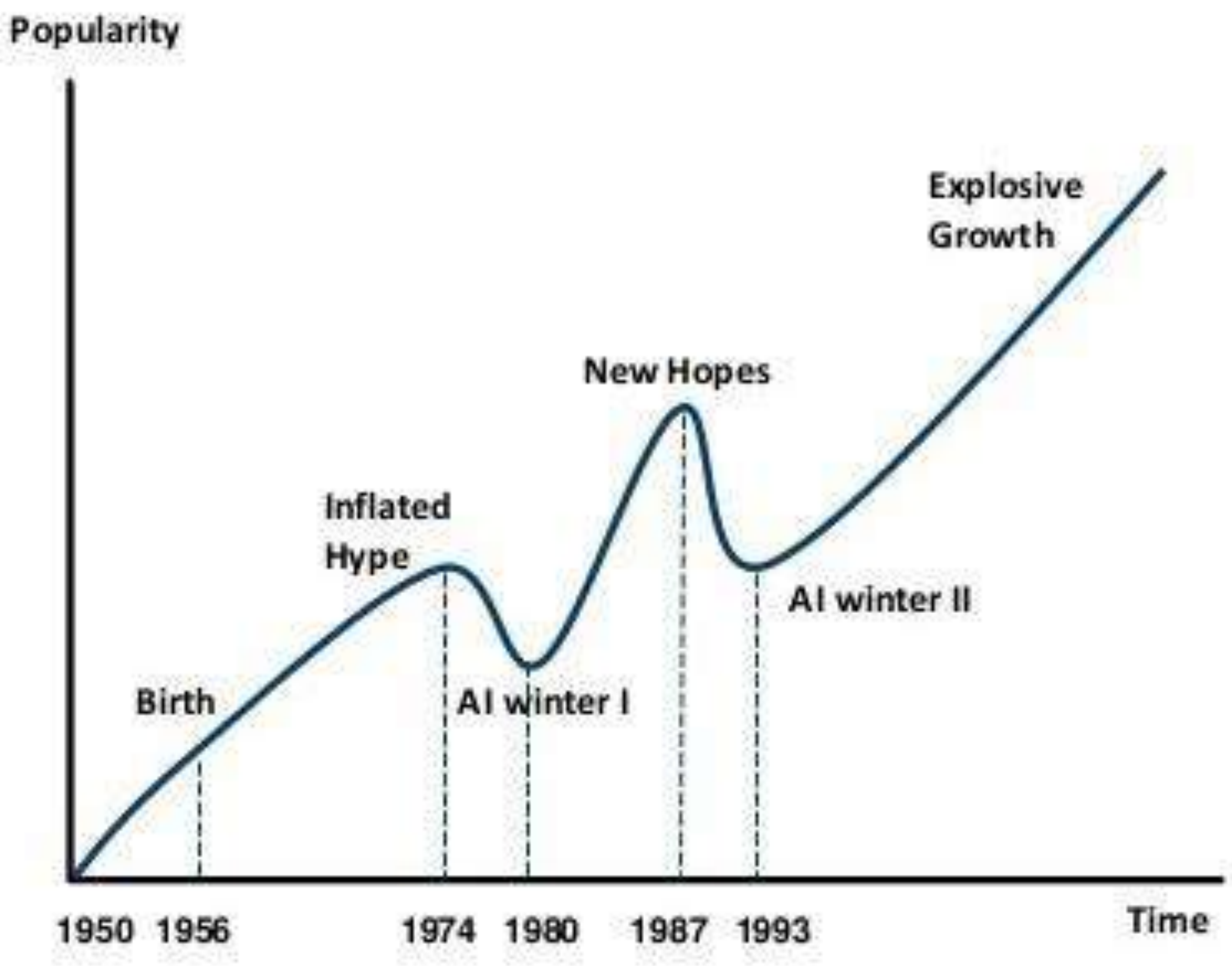


Hype Cycle for Emerging Technologies, 2023

Source: Gartner

AI – Winters and Summers

AI HAS A LONG HISTORY OF BEING "THE NEXT BIG THING"...

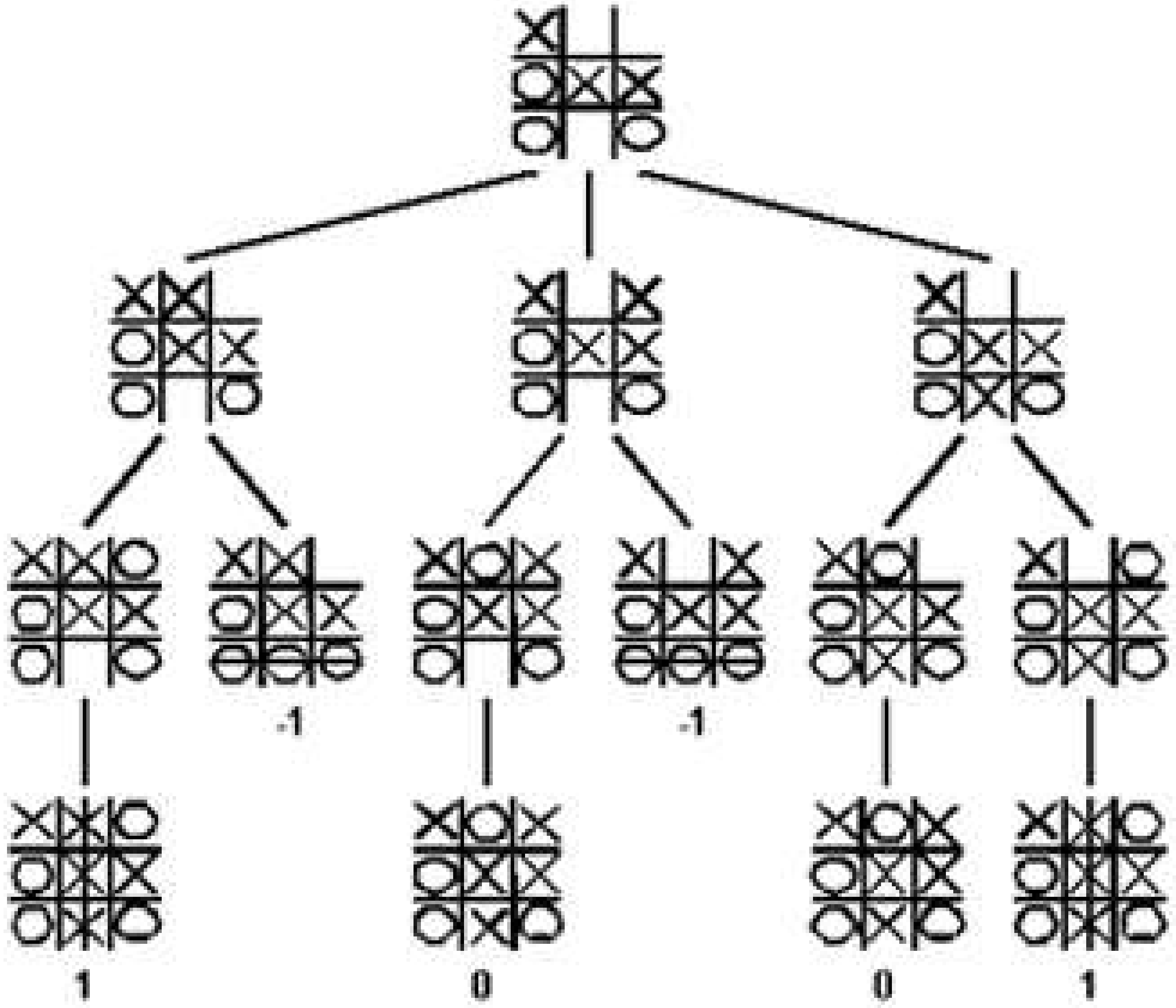


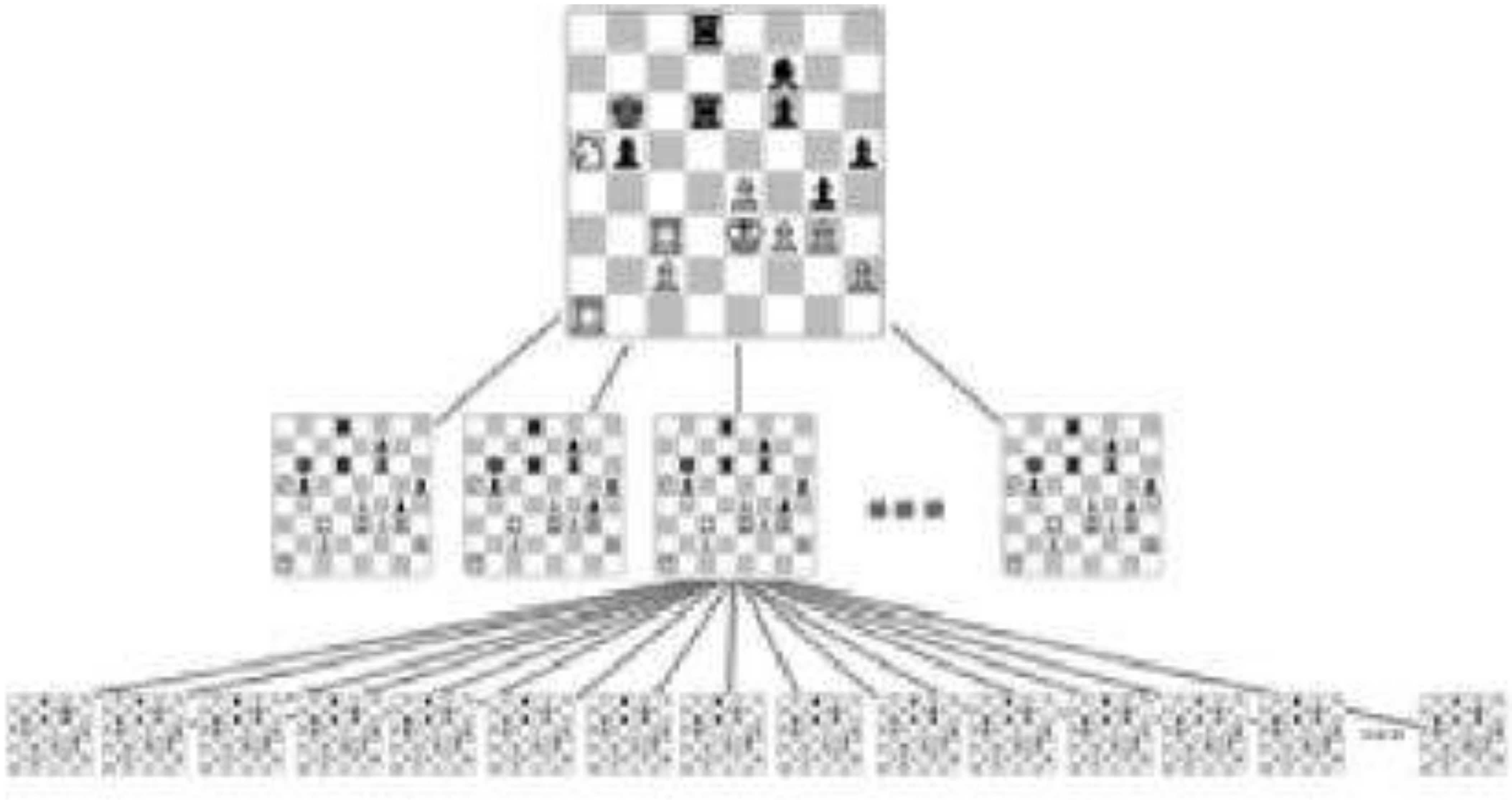
Timeline of AI Development

- **1950s-1960s**: First AI boom - the age of reasoning, prototype AI developed
- **1970s**: AI winter I
- **1980s-1990s**: Second AI boom: the age of Knowledge representation (appearance of expert systems capable of reproducing human decision-making)
- **1990s**: AI winter II
- **1997**: Deep Blue beats Gary Kasparov
- **2006**: University of Toronto develops Deep Learning
- **2011**: IBM's Watson won Jeopardy
- **2016**: Go software based on Deep Learning beats world's champions

AI –latest (ML) summer

Search trees

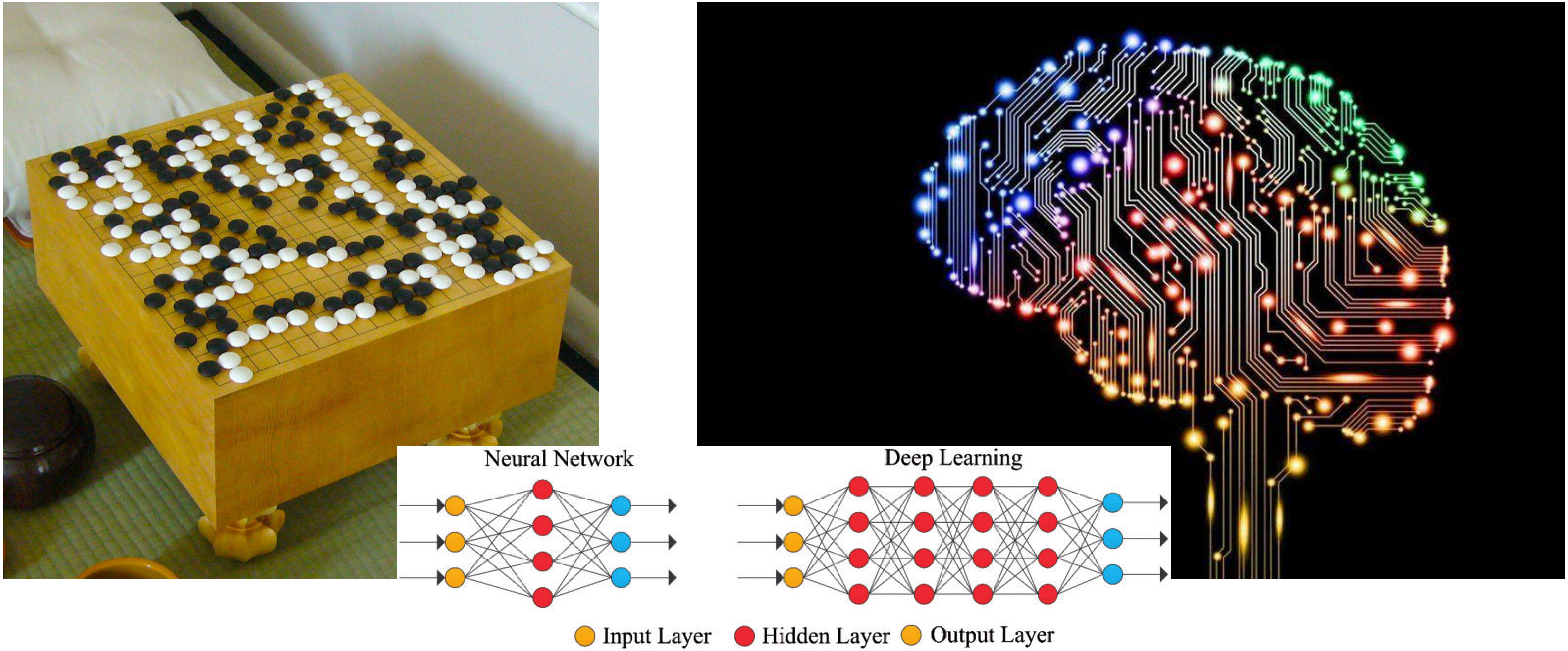




10 years later: Watson

Requires knowledge!

Deep learning & GO



Google Deep Mind: 2016

ChatGPT – 2022 & 2023

hogeschool vives

- Deep learning based

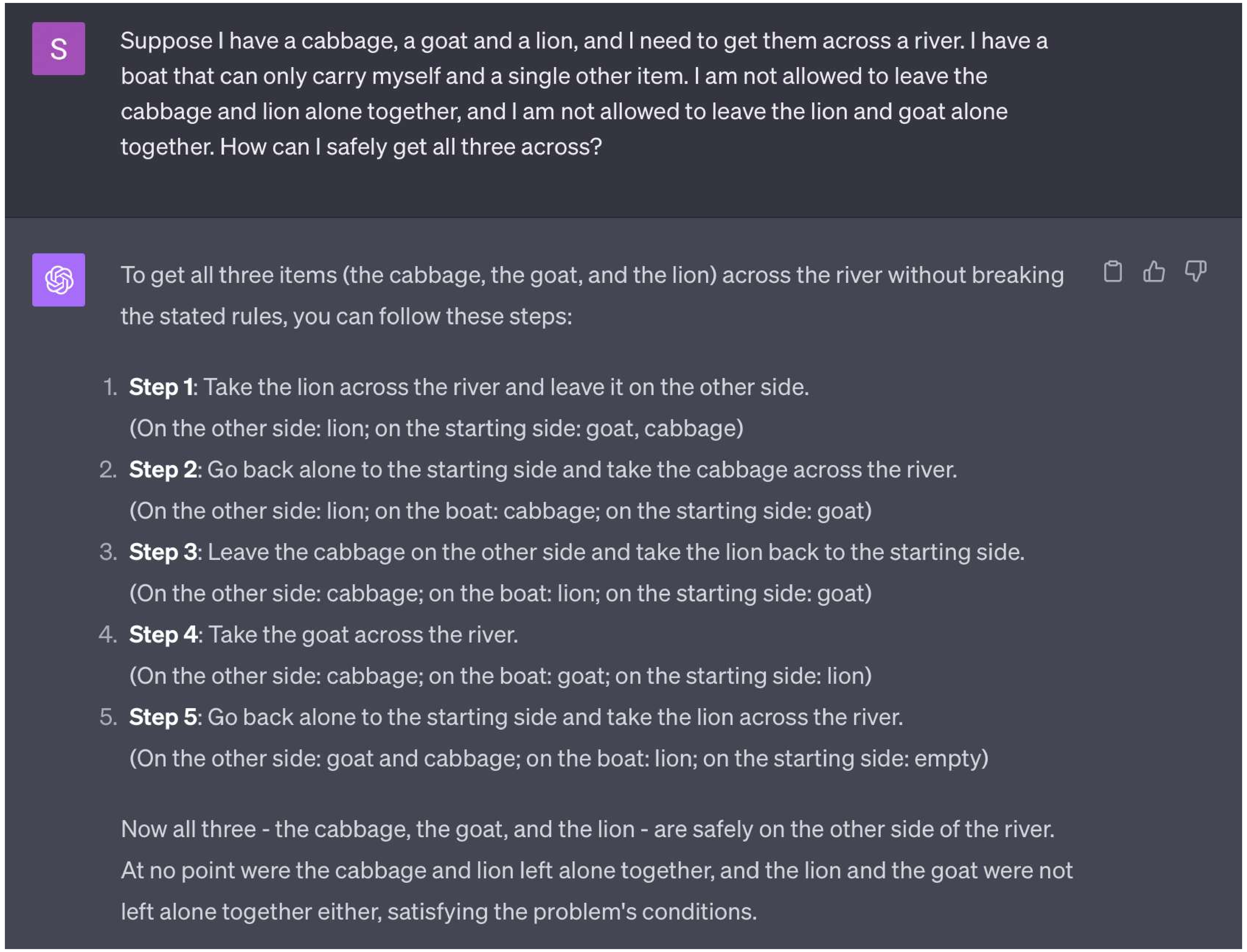
- New model architecture: transformers

- Language model – does NOT know anything!

- Predicts the probability that the next word is X given some previous words

ChatGPT – 2022 & 2023

- The model has several flaws:
  - Bias
  - Racism
  - Wrong answers
  - Solved by adding rules… → degrades to an expert system, a lot of manual programming required!
- The model DOES NOT learn immediately!
- Has NO logic!
- May hallucinate!



S

Suppose I have a cabbage, a goat and a lion, and I need to get them across a river. I have a boat that can only carry myself and a single other item. I am not allowed to leave the cabbage and lion alone together, and I am not allowed to leave the lion and goat alone together. How can I safely get all three across?

To get all three items (the cabbage, the goat, and the lion) across the river without breaking the stated rules, you can follow these steps:

1. **Step 1**: Take the lion across the river and leave it on the other side. (On the other side: lion; on the starting side: goat, cabbage)
2. **Step 2**: Go back alone to the starting side and take the cabbage across the river. (On the other side: lion; on the boat: cabbage; on the starting side: goat)
3. **Step 3**: Leave the cabbage on the other side and take the lion back to the starting side. (On the other side: cabbage; on the boat: lion; on the starting side: goat)
4. **Step 4**: Take the goat across the river. (On the other side: cabbage; on the boat: goat; on the starting side: lion)
5. **Step 5**: Go back alone to the starting side and take the lion across the river. (On the other side: goat and cabbage; on the boat: lion; on the starting side: empty)

Now all three - the cabbage, the goat, and the lion - are safely on the other side of the river. At no point were the cabbage and lion left alone together, and the lion and the goat were not left alone together either, satisfying the problem's conditions.

About intelligence...

- When should we consider a program intelligent?

- When do we consider a (creative) activity of humans to require intelligence?

- Default answers: never? always?

- Most used definition of artificial intelligence:
  - If a task performed by humans requires intelligence, and the software/device is able to perform this task, then the software/device is an artifical intelligence system.

Does numeric computation require intelligence?

- For humans?
  - Calculation: 3921,56 x 73,13 = ?

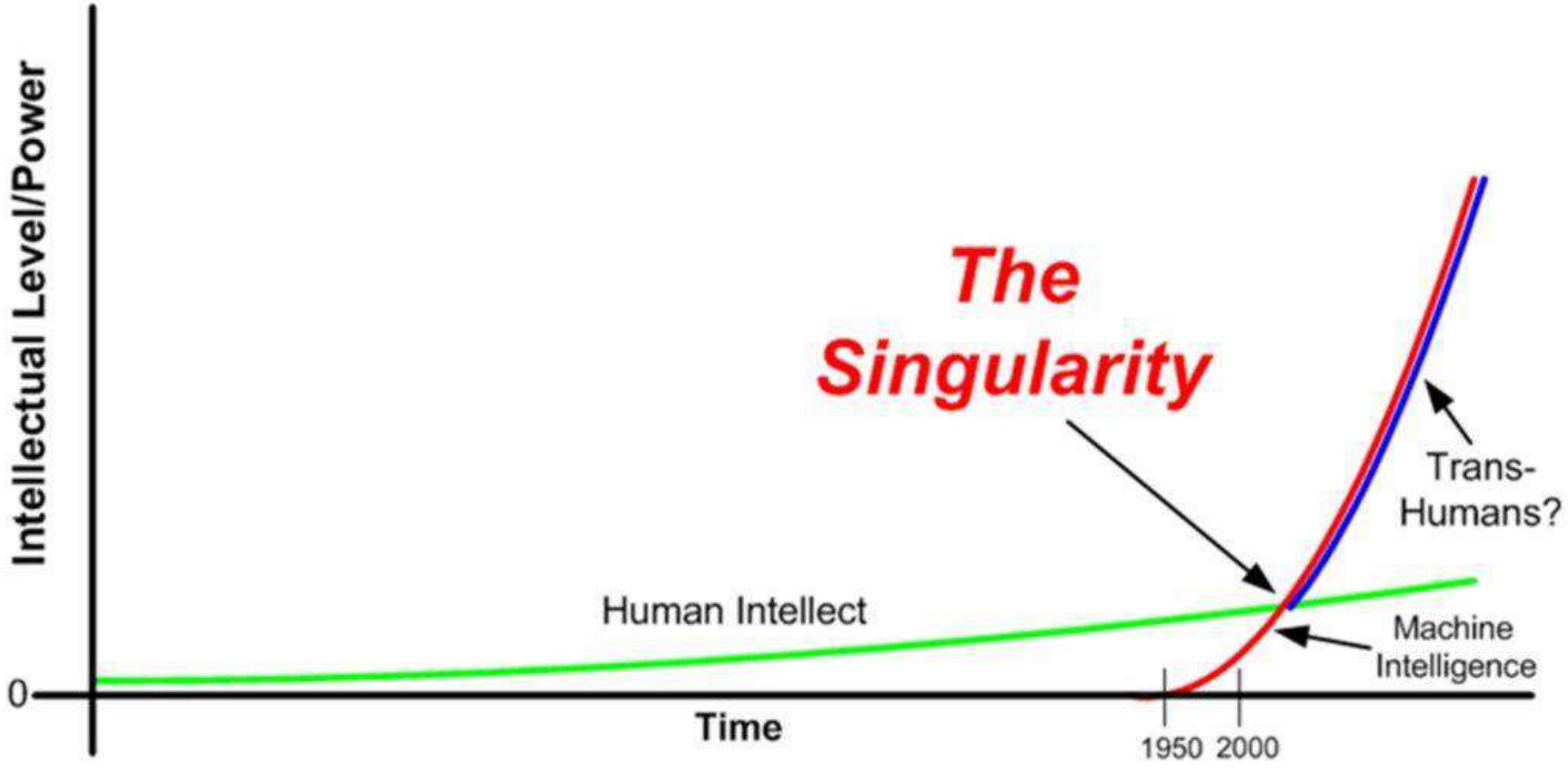
- For computers?

- When do we consider a program “intelligent”?
  - And when will computers ‘overrule’ us?

To situate the question: two aims

- **Long term aim:**
  - - develop systems that achieve a level of “intelligence” similar/comparable/better? than that of humans
  - - not achievable in the next 20 to 30 years
- **Short term aim:**
  - - on specific tasks that seem to require intelligence
  - - develop systems that achieve a “level of intelligence” similar/comparable/better? than that of humans
  - - achieved for very many tasks already: deep blue, data mining, computer vision, ...

Long term: point of singularity



Intermezzo

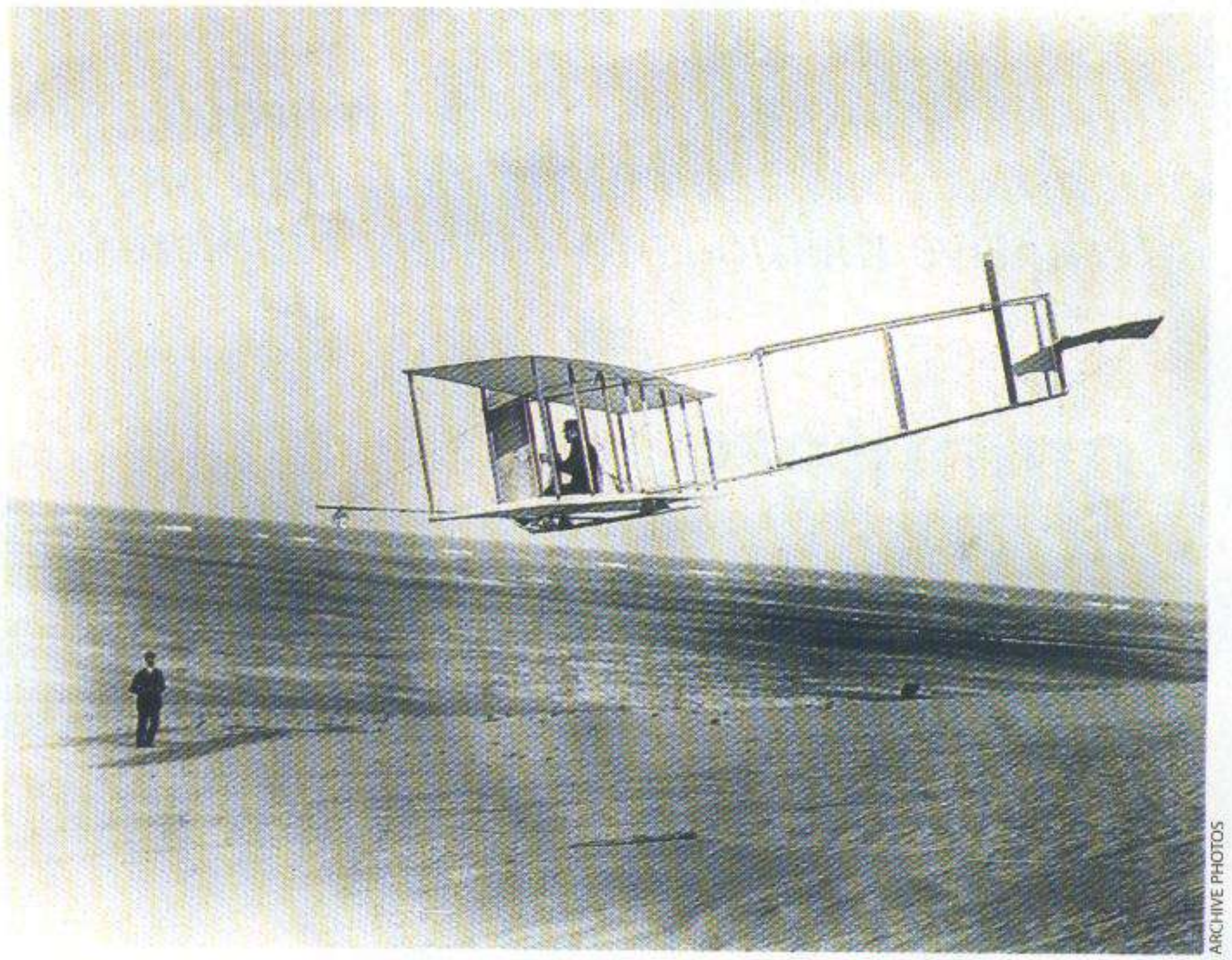
- A stupid question:

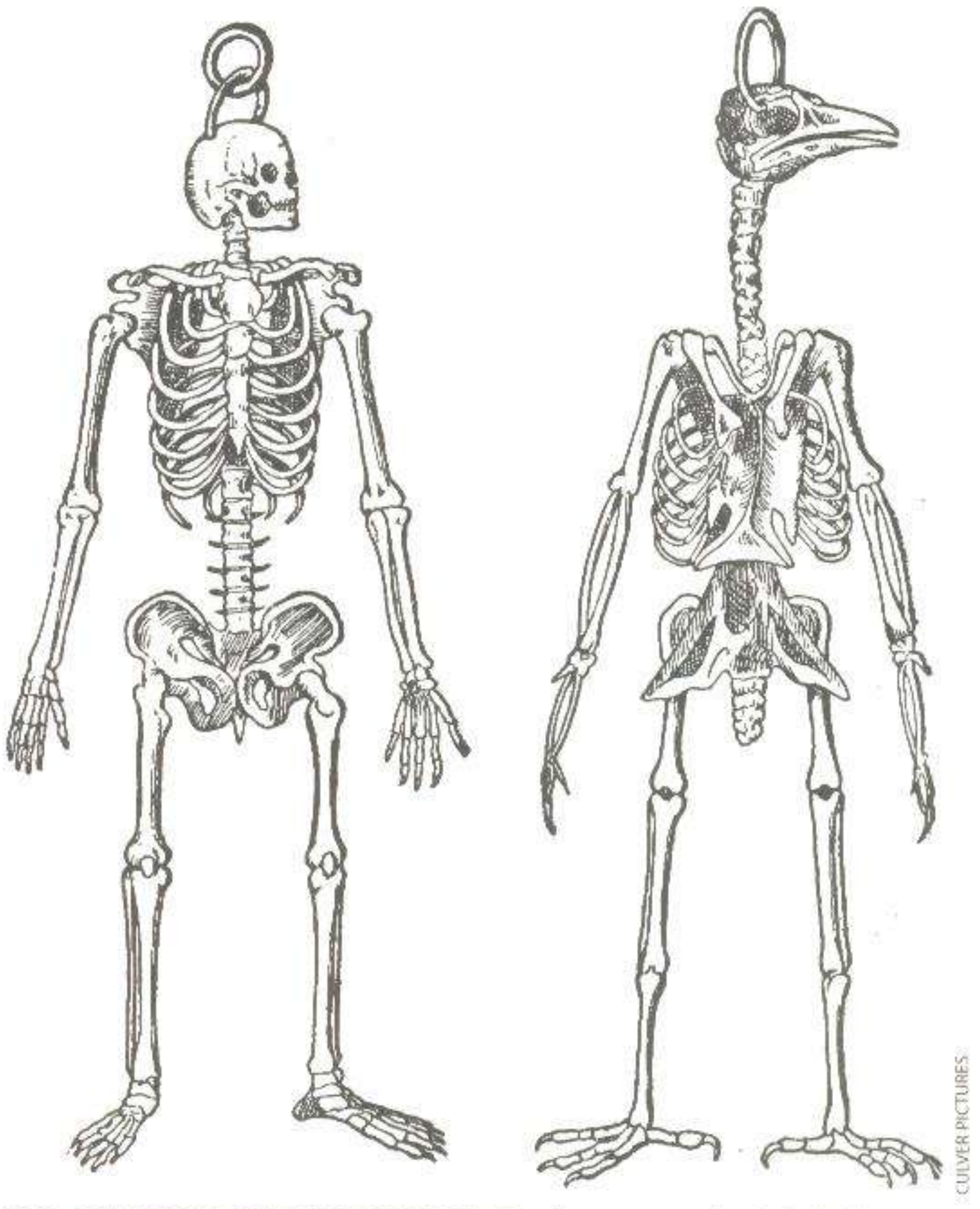
- Can people fly?

Short term: reproduction versus simulation

- We are not to SIMULATING human intelligence

- We are REPRODUCING the effect of intelligence


ARCHIVE PHOTOS

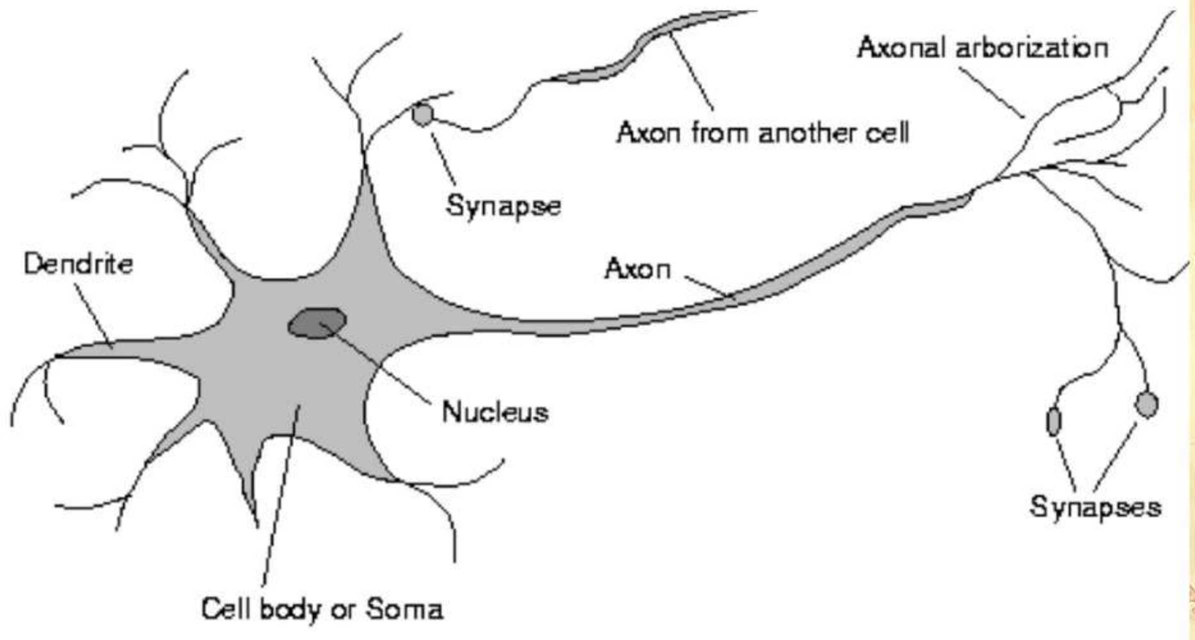

CULVER PICTURES

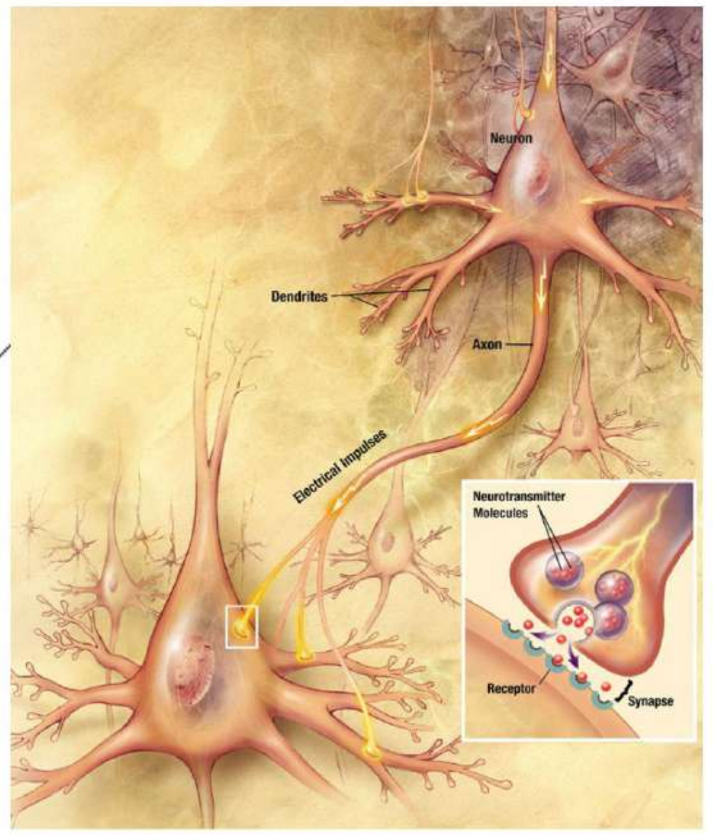
COMPARISON OF SKELETONS of a human and a bird—here taken from a 16th-century manuscript by French naturalist Pierre Belon—examined similarites in anatomy in an attempt to understand how birds can fly.

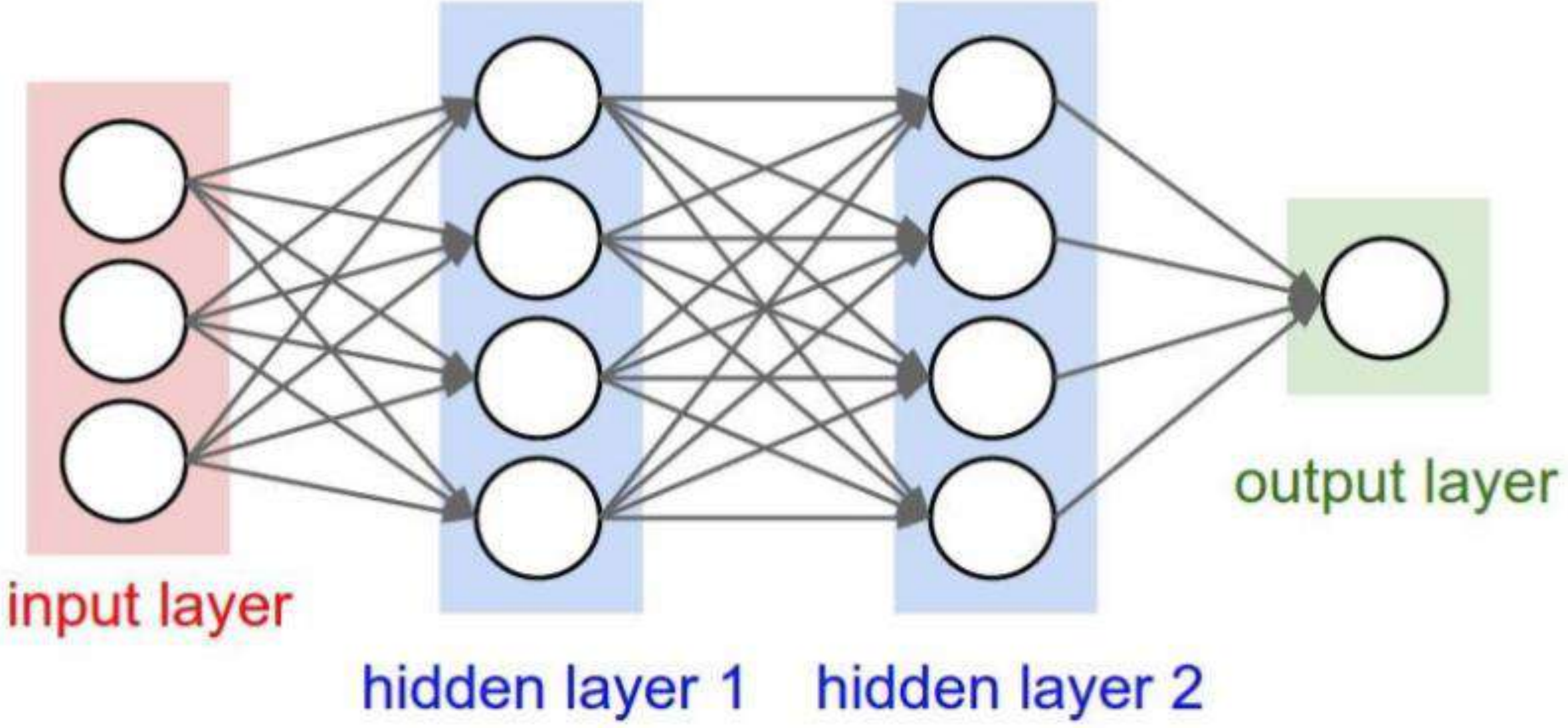
Will AI overrule us?

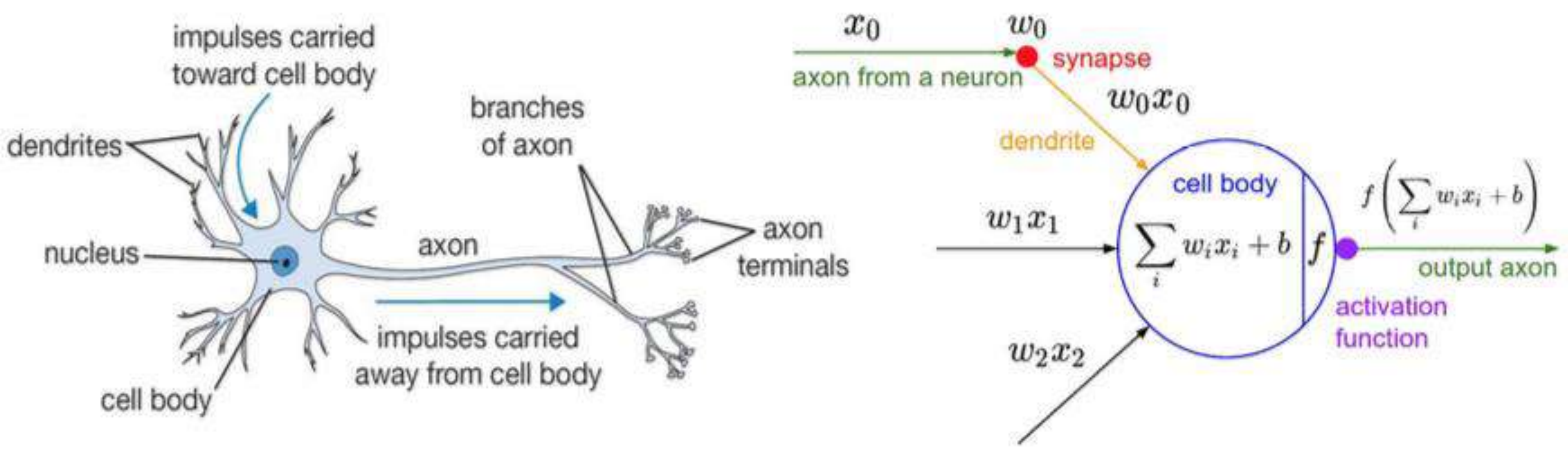
- - Is more or less the question: will deep learning overrule us?

Biology of a Neuron









AI – ML - DL

- **- Artificial Intelligence** (AI):

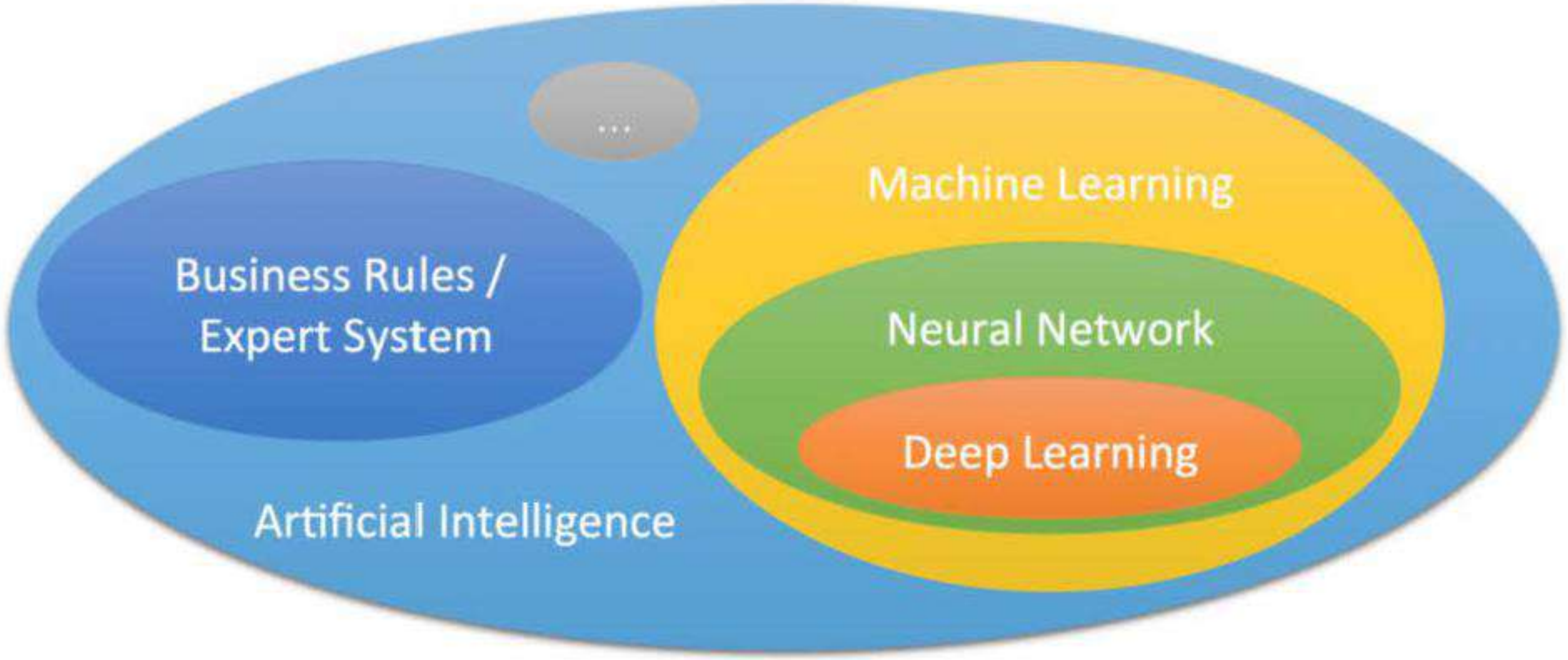
  “The set of all tasks in which a computer can make decisions.”

- **- Machine Learning** (ML):

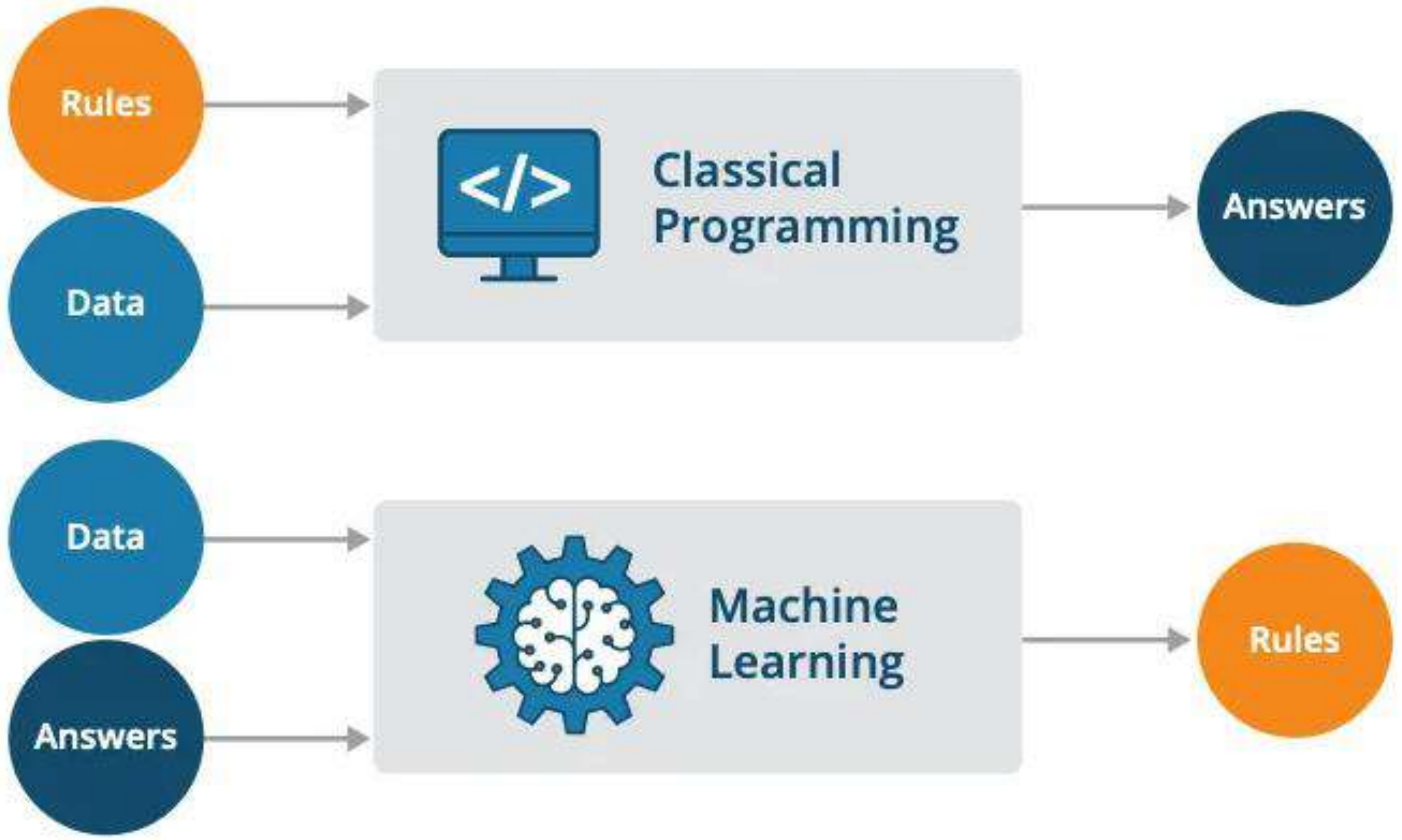
  “The set of all tasks in which a computer can make decisions *based on data*.”

- **- Deep Learning** (DL):

  “The field of machine learning that uses certain objects called *neural networks*.”



AI – ML - DL



Rules

Data

Classical Programming

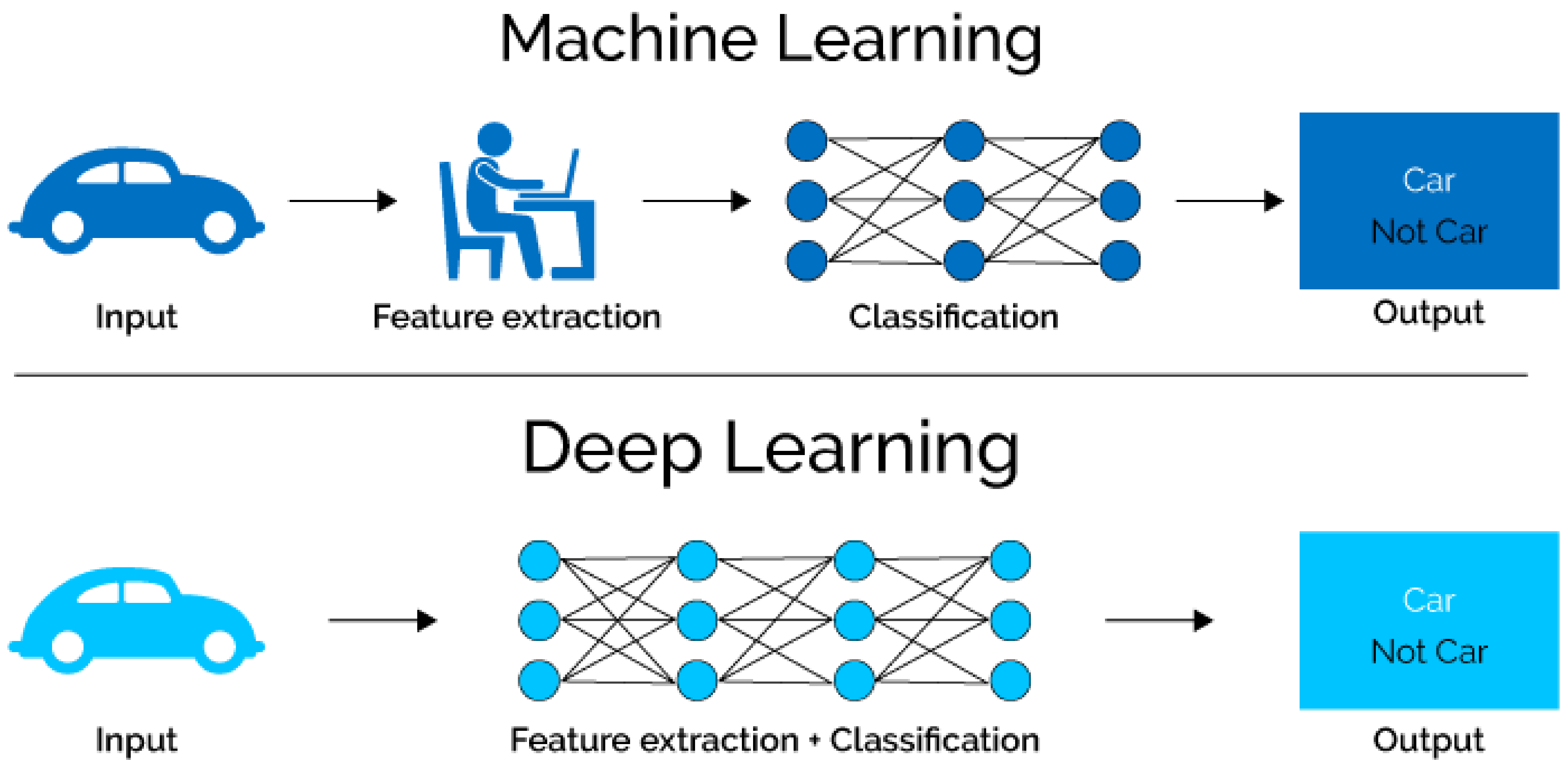
Answers

Data

Answers

Machine Learning

Rules



Machine Learning

Input

Feature extraction

Classification

Car

Not Car

Output

Deep Learning

Input

Feature extraction + Classification

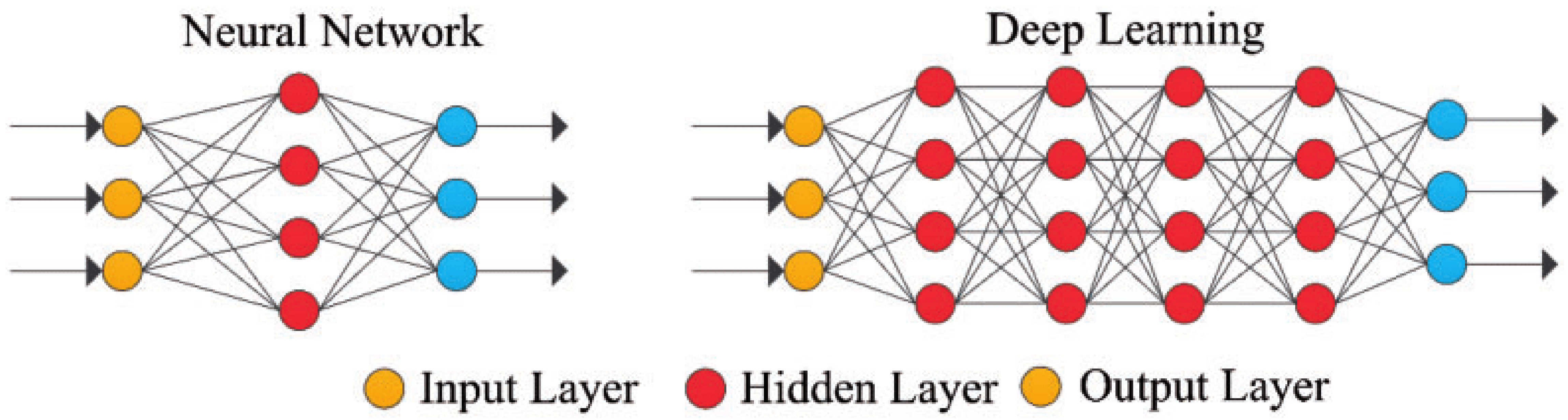
Car

Not Car

Output

- Issue with ML:

- Not always interpretable
- Needs (a lot of) data



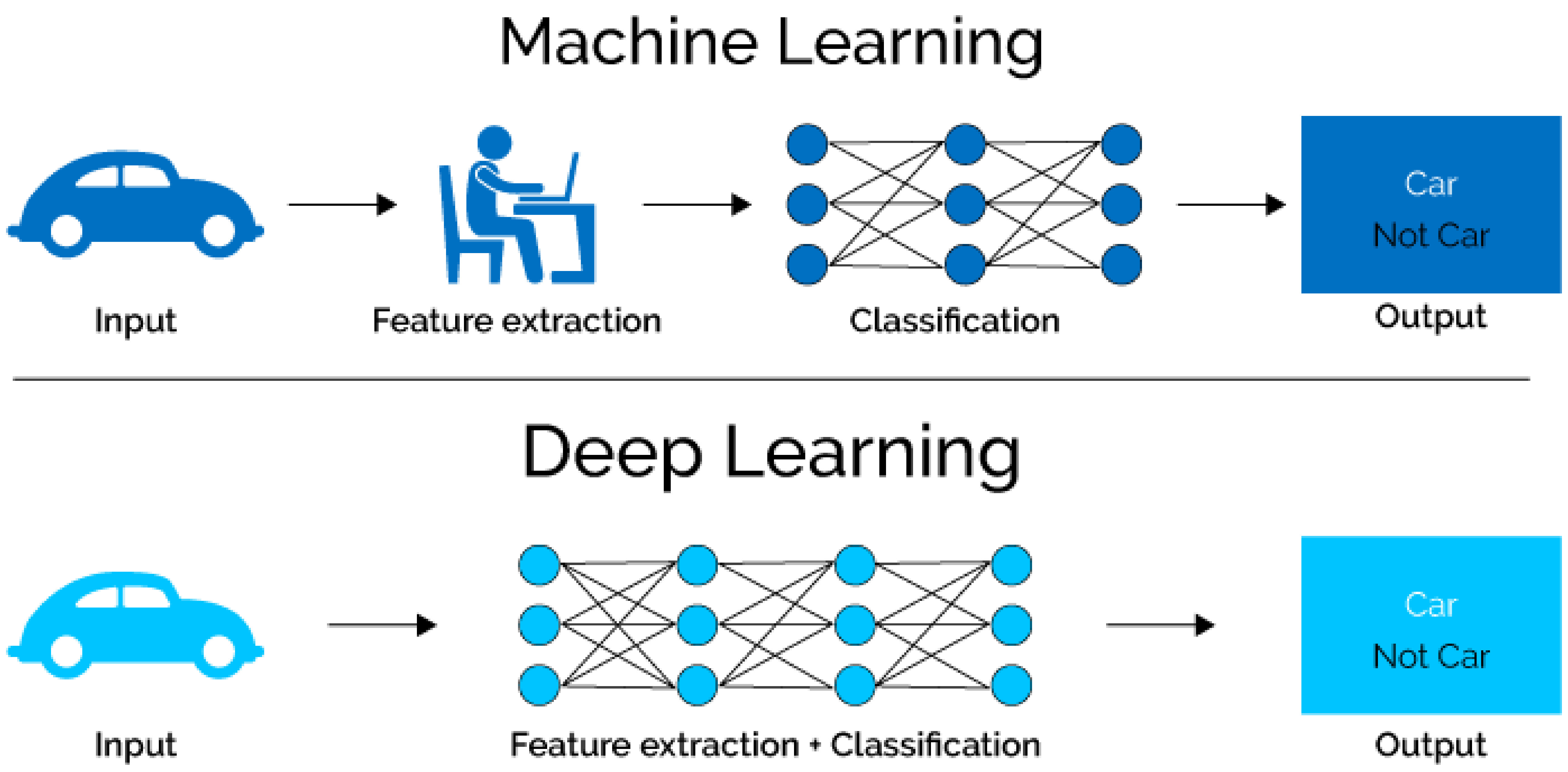
Neural Network

Deep Learning

● Input Layer ● Hidden Layer ● Output Layer

What is deep learning?

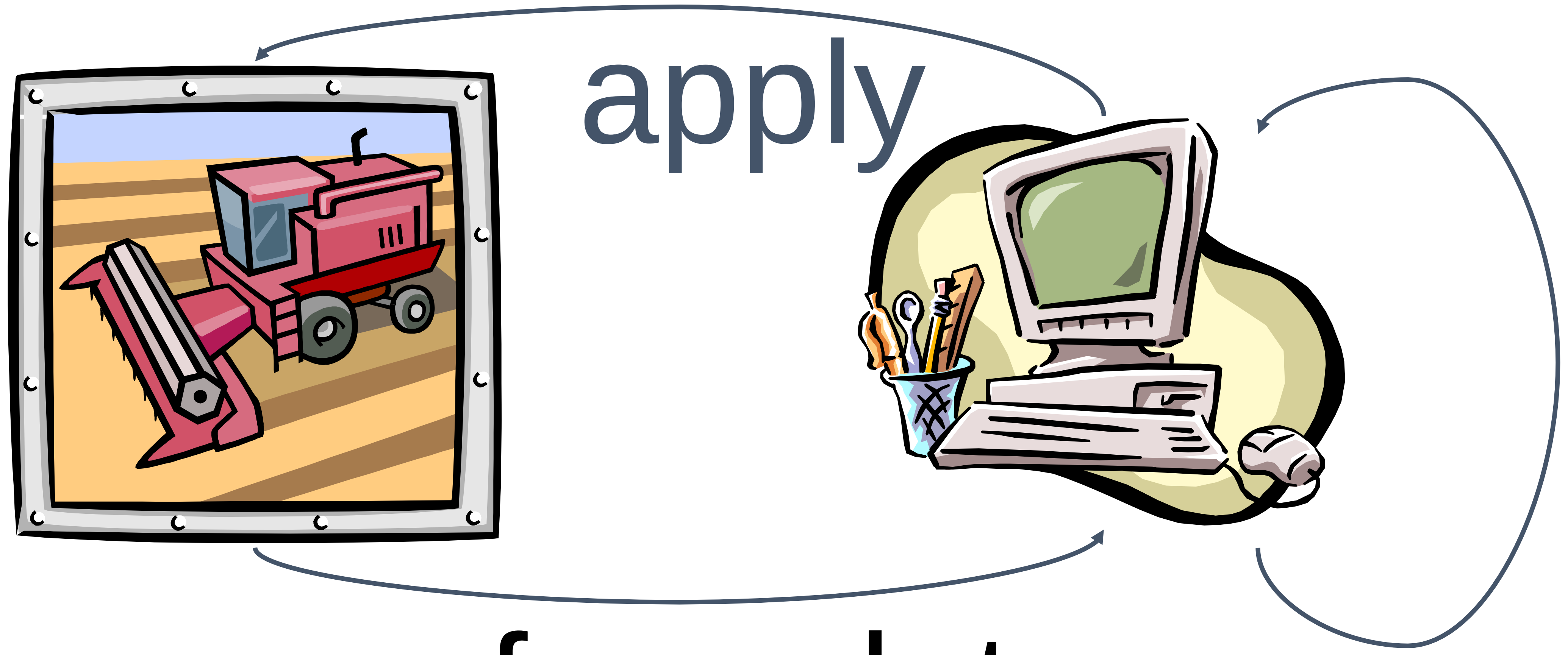
- Subfield of ML for **learning representations** of data.
- Exceptional effective at **learning patterns**.
- Utilizes learning algorithms that derive **meaning** out of data by using a **hierarchy of multiple layers** that mimic the **neural networks of our brain**.
- If you provide the system **tons of information**, it begins to understand it and respond in useful ways.



https://www.xenonstack.com/blog/static/public/uploads/media/machine-learning-vs-deep-learning.png

What is AI? For me…

- Computers/algorithms making decisions/predictions in real-world problems



apply

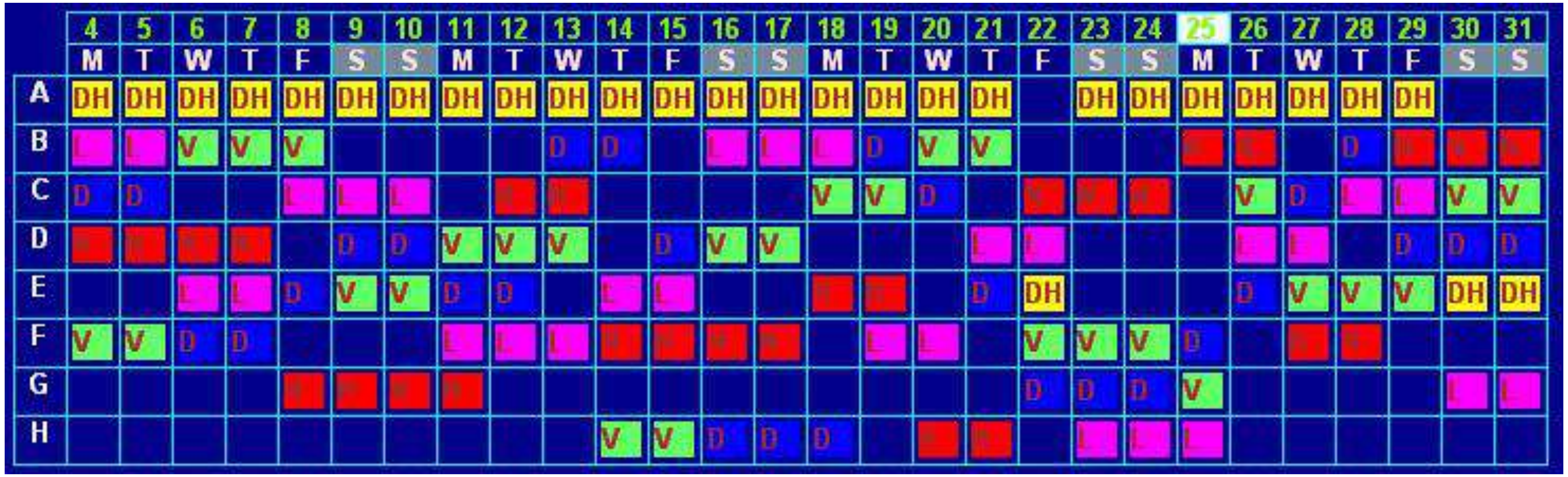
solve

formulate

AI – example – employee rostering

- Problem:

- assignments of shifts to employees with a certain qualification
- considering a large number of constraints and preferences



	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
A	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH	DH		DH	DH	DH	DH	DH	DH	DH		
B	L	L	V	V	V					D	D		L	L	L	D	V	V				N	N		D	N	N	N
C	D	D			L	L	L		N	N					V	V	D		N	N	N		V	D	L	L	V	V
D	N	N	N	N		D	D	V	V	V		D	V	V				L	L				L	L		D	D	D
E			L	L	D	V	V	D	D		L	L			N	N		D	DH				D	V	V	V	DH	DH
F	V	V	D	D				L	L	L	N	N	N	N		L	L		V	V	V	D		N	N			
G					N	N	N	N											D	D	D	V					L	L
H											V	V	D	D	D		N	N		L	L	L						

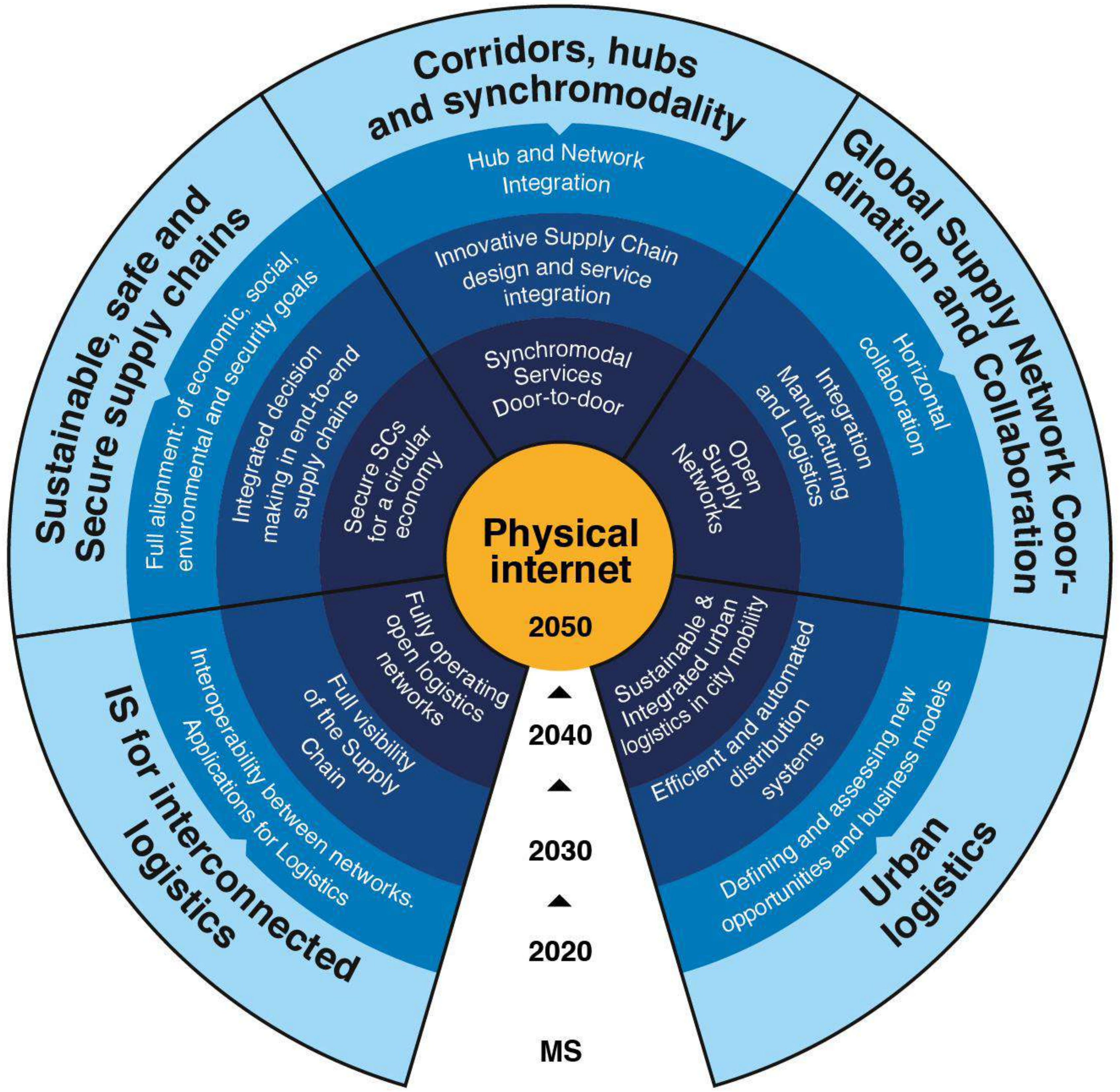
AI – example - employee rostering

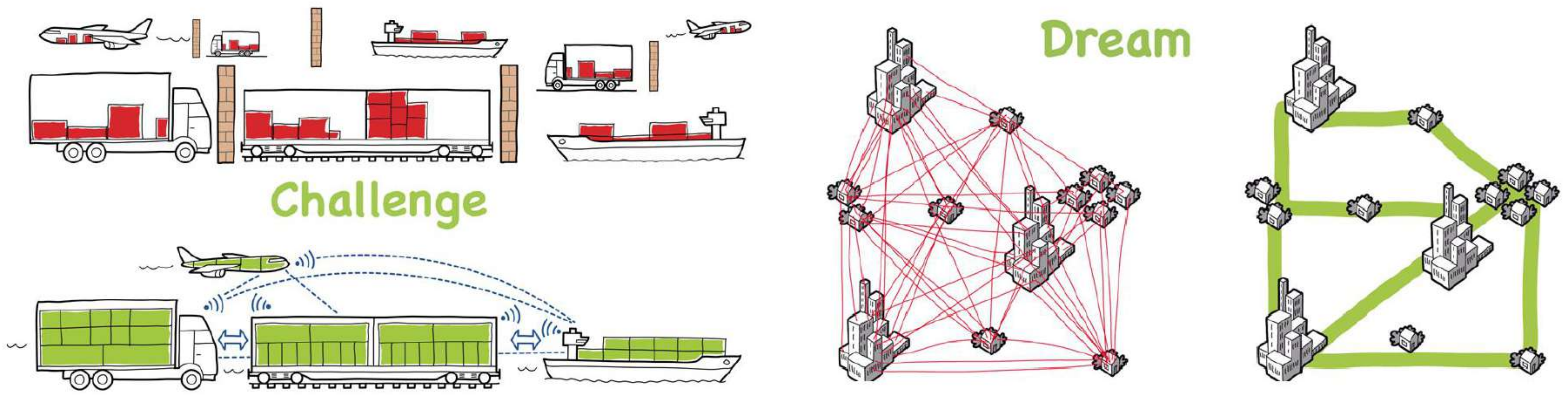
- Small example:
  - 4 shifts to schedule
  - 10 possible eployees
  - 7 days

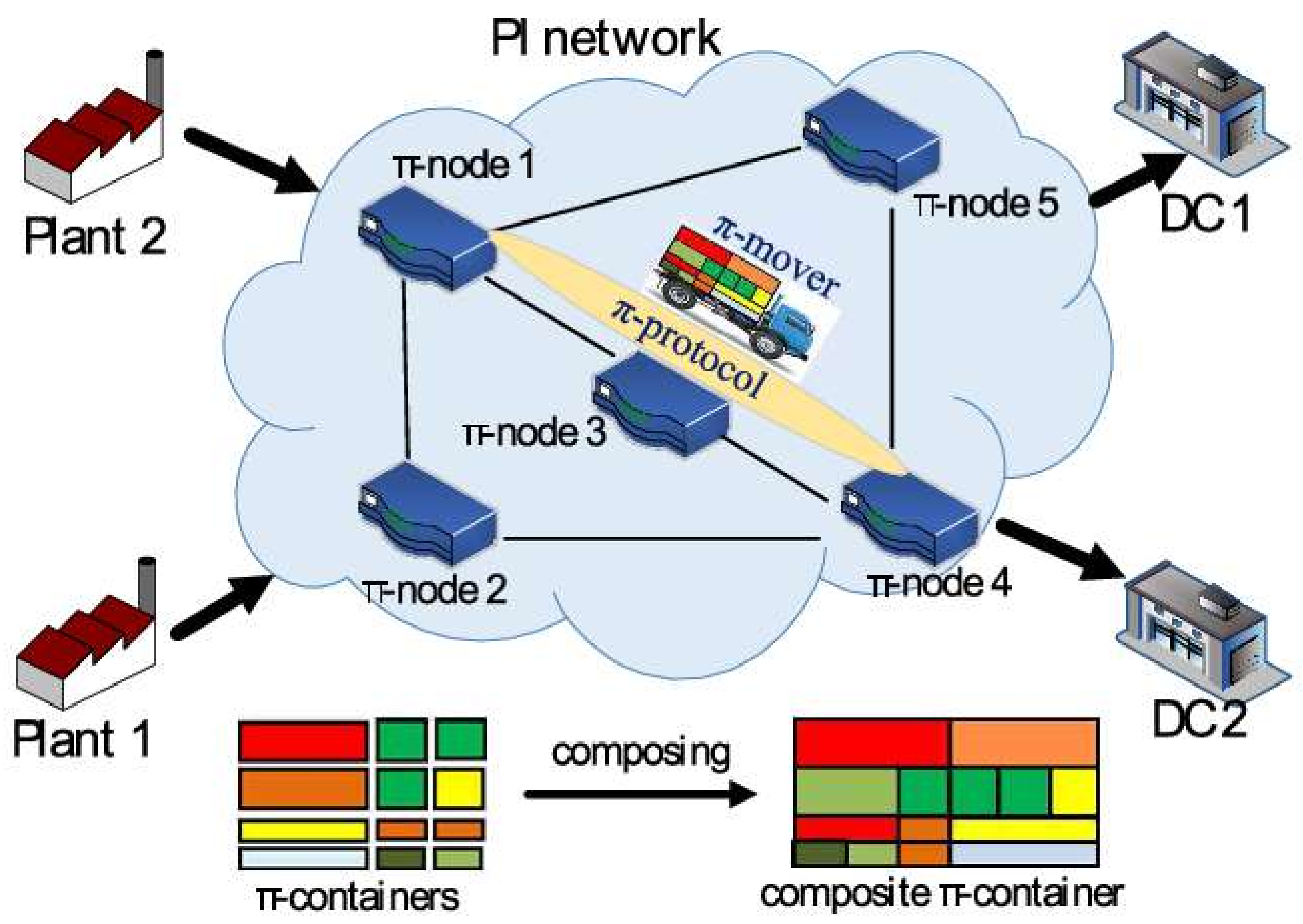
- Number of variations on 1 day = 5040
  - For 1 week: number of possible rosters = 5040^7 = 8,260641125390352e25
  - For 1 month: number of possible rosters = 5040^{30} = 1,182813011613388e111

AI – synchromodal transport

- Physical internet


Corridors, hubs and synchromodality
Hub and Network Integration
Innovative Supply Chain design and service integration
Synchromodal Services Door-to-door
Global Supply Network Coordination and Collaboration
Horizontal collaboration
Integration Manufacturing and Logistics
Open Supply Networks
Urban logistics
Defining and assessing new opportunities and business models
Efficient and automated distribution systems
Sustainable & Integrated urban logistics in city mobility
IS for interconnected logistics
Interoperability between networks. Applications for Logistics
Full visibility of the Supply Chain
Fully operating open logistics networks
Sustainable, safe and Secure supply chains
Full alignment: of economic, social, environmental and security goals
Integrated decision making in end-to-end supply chains
Secure SCs for a circular economy
Physical internet
2050
2040
2030
2020
MS





AI – combinatorial optimisation

- Combinatorial optimisation is a field in mathematics and computer science

- The goal is to **find the best solution from a finite set of possible solutions** based on **specific criteria or constraints**.

- By systematically **considering each one** and selecting the one that meets the defined optimisation criteria.

AI – combinatorial optimisation

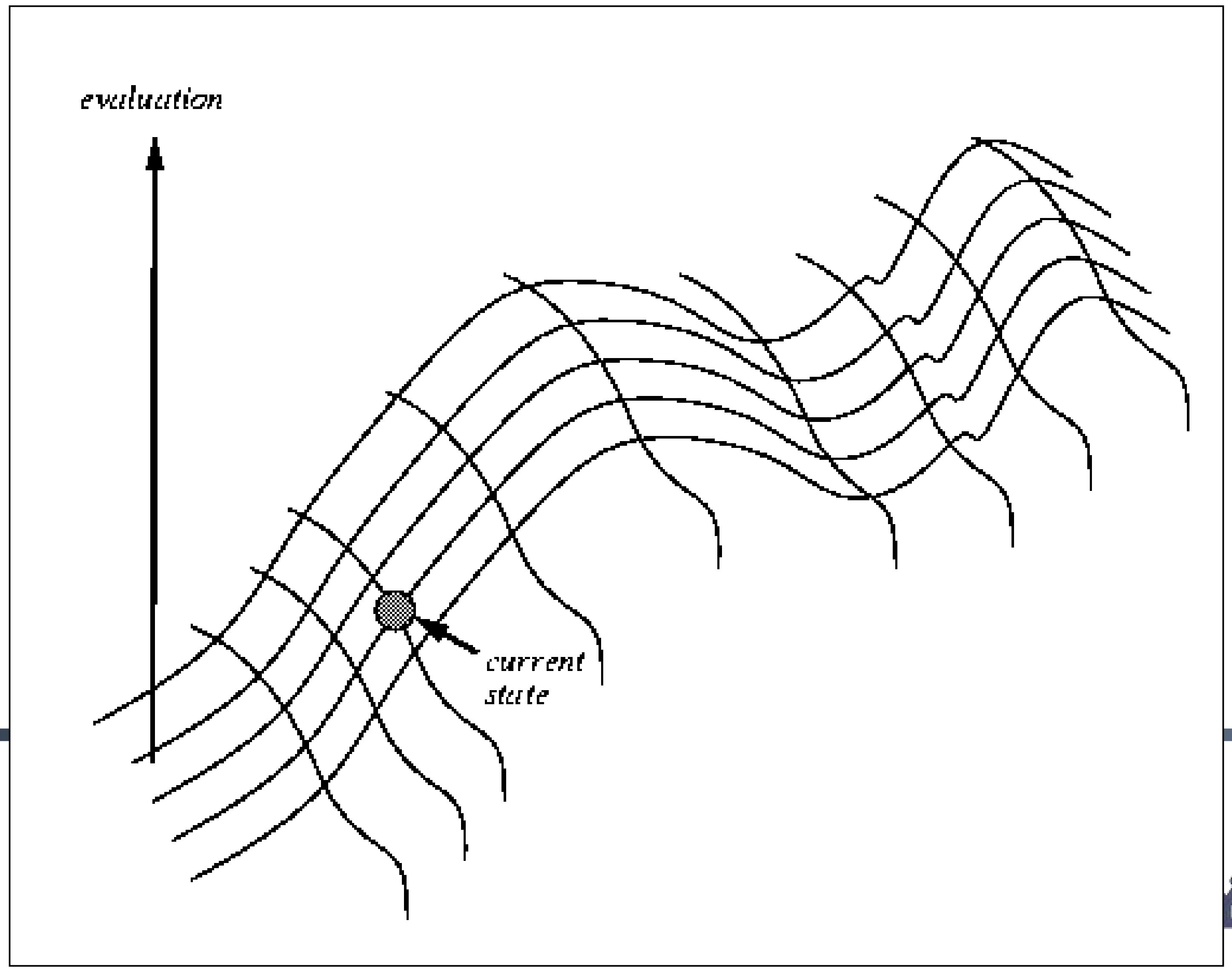
- **How to solve?**
- Integer programming:
  - Modelling is important!
  - Excellent commercial solvers:
    - CPLEX, GUROBI, …
    - can be expensive
  - Great quality open-source solvers:
    - COIN-OR, …
    - Not always stable
  - In general:
    - Can take a lot of time
    - Calculation power
    - Often intractable!

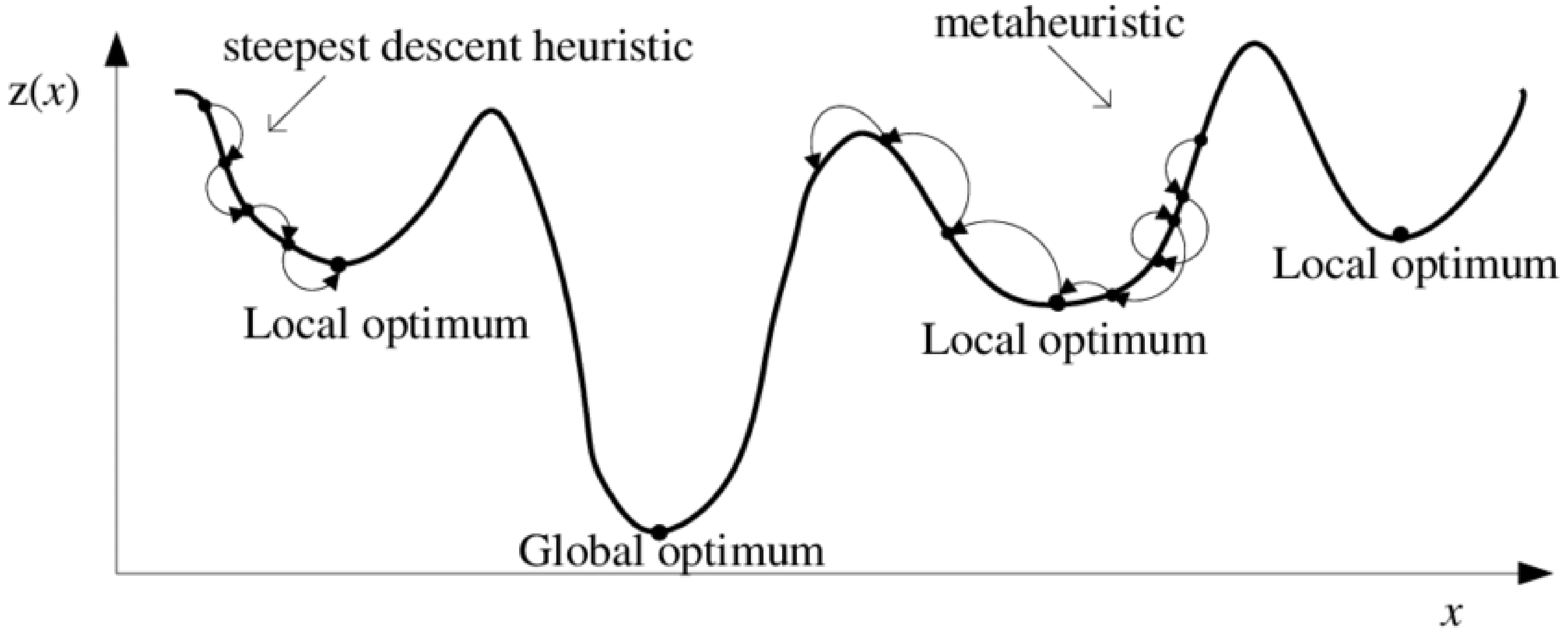
AI - combinatorial optimisation

- **<u>How to solve?</u>**

- Use metaheuristics:
  - Generate a start solution
  - “Improve” the solution, by making “specific changes” to the solution
  - Stop if no better solution can be found

- Often used:
  - Hill climbing
  - Steepest descent
  - Tabu Search
  - Genetic algorithms
  - ALNS





Tabu search

- Definitions

  - Neighbourhood solution: a solution which must exist in a set of feasible solutions, and which is not in the tabu list.

  - Move: change the current solution to its neighbourhood solution.

  - Tabu List: a short-term memory which records the solutions that have been visited in the recent past. In this way, we can avoid repeating search. In general, tabu list has a fixed size to memorize, and it follows FIFO to maintain the list.

Tabu search

- Some stopping criteria
  - Maximum iterative numbers
  - Maximum times which counts when object function’s value doesn’t improve
  - The longest default execution time of CPU
  - When object function’s output is acceptable

Tabu search

- Example: tabu search for Traveling Salesman Problem

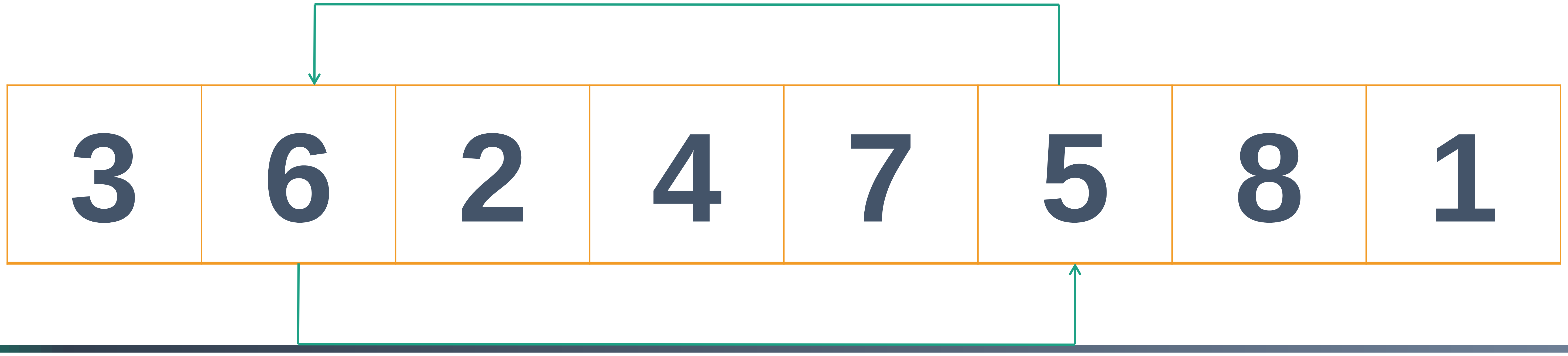
- A feasible solution is represented as a sequence of nodes, each node appearing only once, and in the order it is visited. For convenience, node 1 is fixed as start and ending node
- Initial Solution
  - A good feasible, yet not-optimal, solution to the TSP can be found quickly using a greedy approach.
  - Starting with the first node in the tour, find the nearest node.
  - Each time find the nearest unvisited node from the current node until all the nodes are visited.

3	6	2	4	7	5	8	1

Tabu search

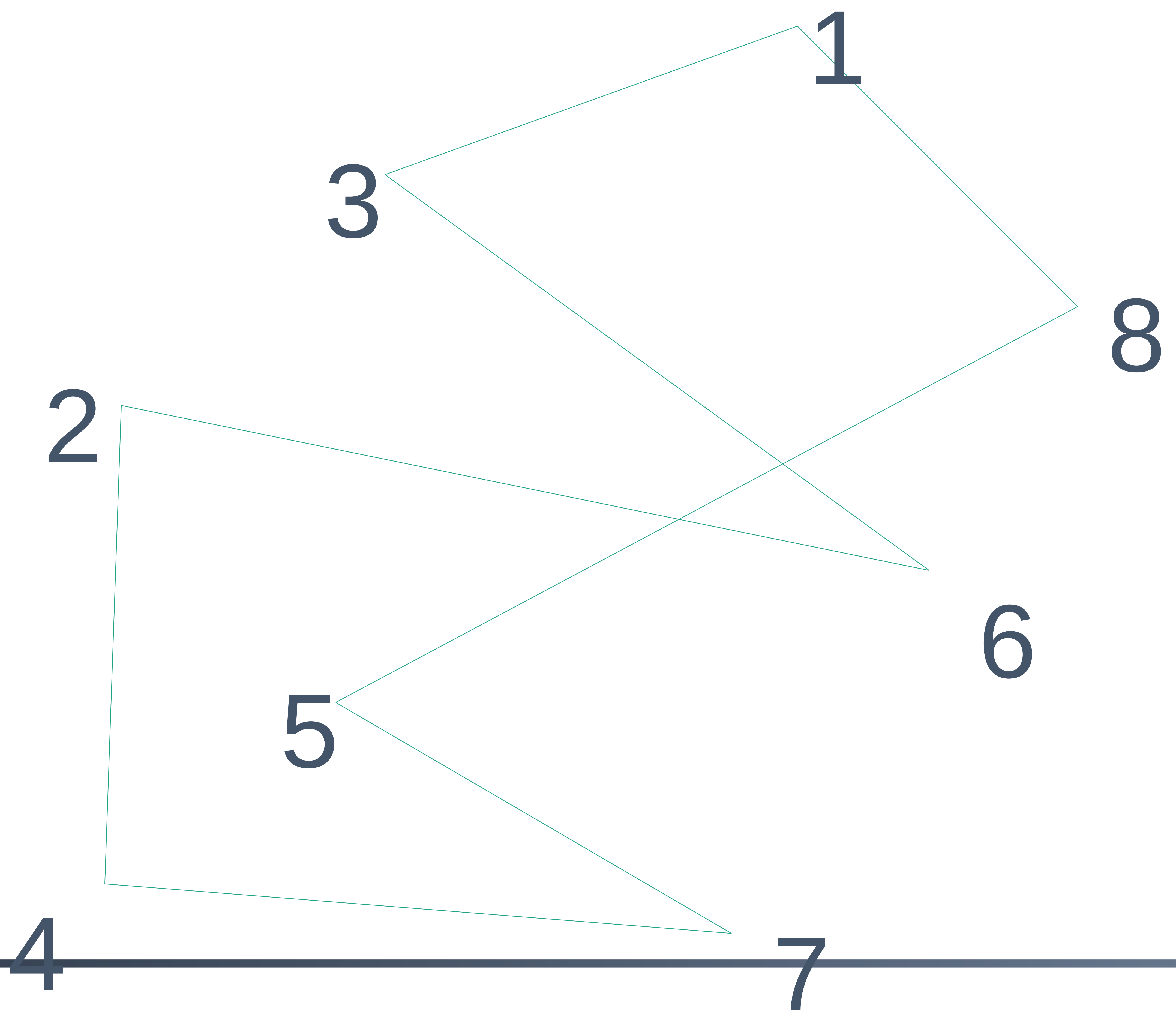
- Neighborhood solution

- A neighborhood solution to a given solution is defined as any other solution that is obtained by a pair wise exchange of any two nodes in the solution.
- If we fix node 1 as the start and the end node, for a problem of N nodes, there are Cn-12 such neighborhoods to a given solution.

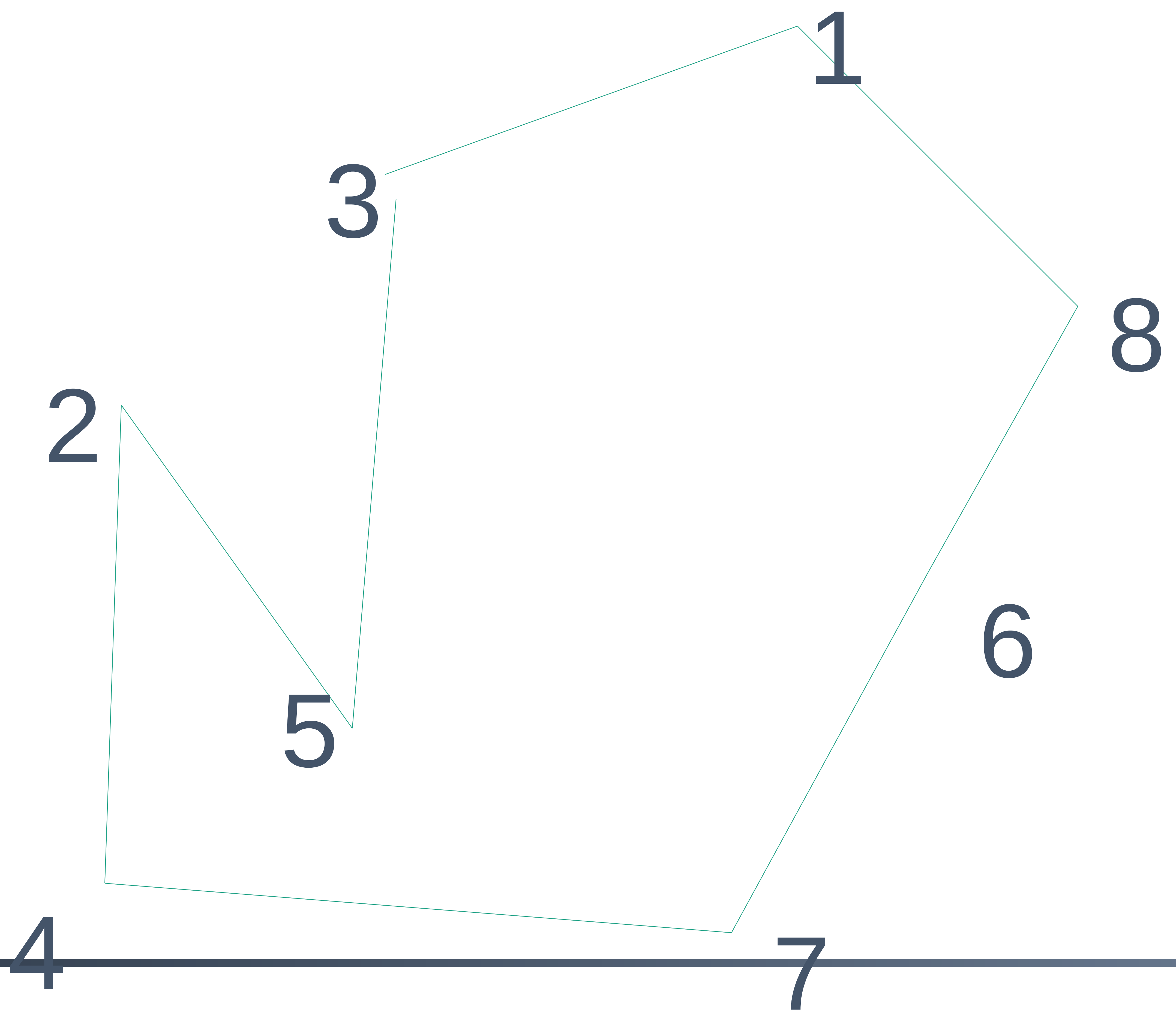


3	6	2	4	7	5	8	1

Tabu search



Tabu search



Tabu search

- Tabu List
- Initially, it is empty
- the attribute stored in the Tabu list is a pair of nodes that have been exchanged recently.

- Aspiration criteria
- The criterion used for this to happen in the present problem of TSP is to allow a move, even if it is in tabu list, if it results in a solution with an objective value better than that of the current best-known solution.

Tabu search

- Problem!

- Too slow for our problem!

- Reason:

- Neighbourhoods only make local changes with limited impact

- Hard-to-schedule requests are hard to move <-> neighbourhood structure

- “Many iterations needed for small gains”

- …

ALNS

- Solution = ALNS, adaptive large neighbourhood search
- Large = "destroy" and "restore" larger parts of the solution

- 2 phases:
- Remove/destroy-fase
- Repair-fase

- Various remove and repair operators possible

- Good results: able to timetable 100 requests for 6 buses in less than 1 hour!

More applications + techniques

- AI in industry:
  - Early detection of wear in a machine → survival analysis
  - Decide which operator to assign to which machine: recommender system + combinatorial optimisation

- AI in biomedical applications & health:
  - Early detection of kidny failure after ICU → survival analysis
  - Drug discovery based on machine learning
  - Nurse rostering

- AI in education:
  - (Early) detection of dropout based on moodle data
  - Recommendation systems for personalised learning

- AI in news:
  - “Safe” recommendation engignes for news sites

- Questions?

- Contact details:

  Dr. Stefaan Haspeslagh
  Liaison Officer AI, KU Leuven – Vives
  Doorniksesteenweg 145 – 8500 Kortrijk - Belgium
  stefaan.haspeslagh@vives.be
  stefaan.haspeslagh@kuleuven.be