

# Introduction to AI

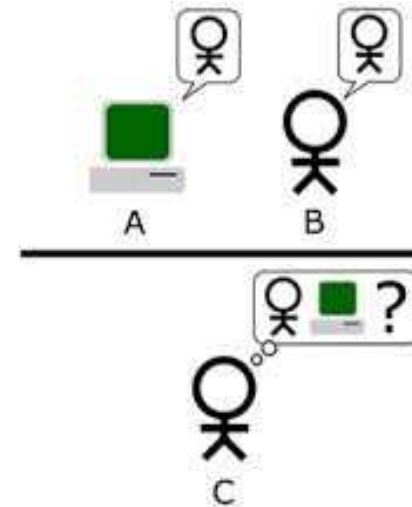
Hwanjo Yu

POSTECH

<http://di.postech.ac.kr/hwanjoyu>

# The Turing Test, 1950

“Can machines think?”



Q: Please write me a sonnet on the subject of the Forth Bridge.

A: Count me out on this one. I never could write poetry.

Q: Add 34957 to 70764.

A: (Pause about 30 seconds and then give as answer) 105621.

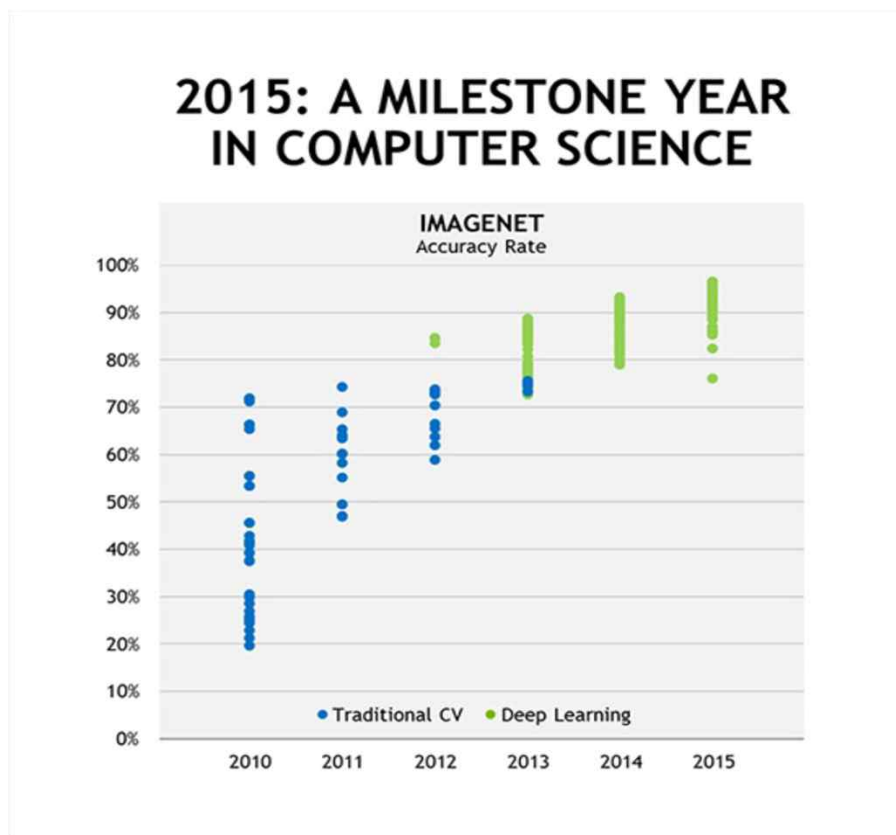
*AI = human-level intelligence?*

*What can and can't computer do better than Human?*

# Computer Vision



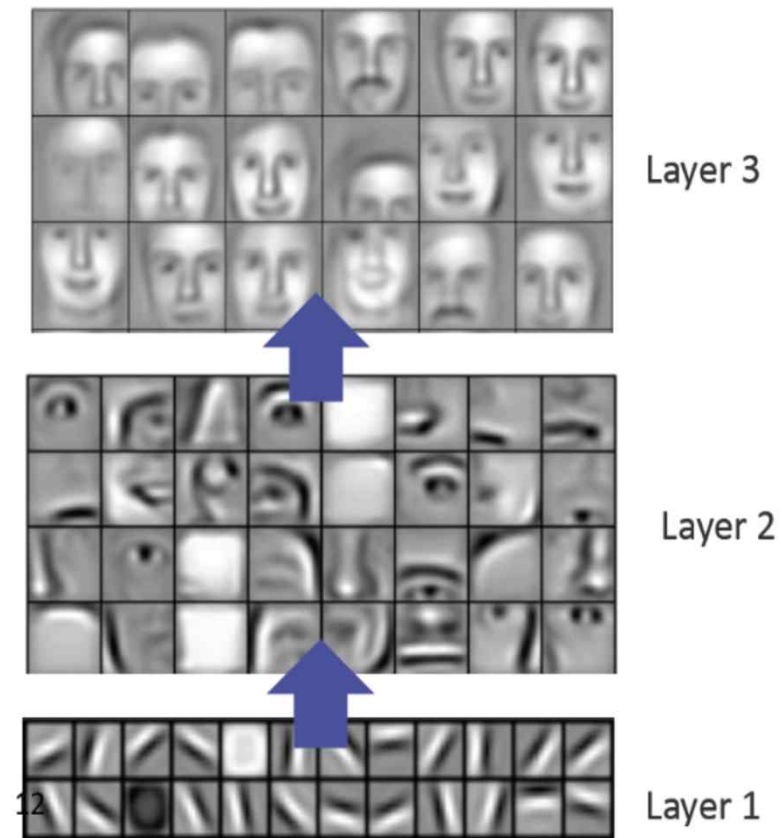
# 2015: A Milestone Year in Computer Vision



- AlexNet (5 convolutional layers + 3 fully connected layers), 2012
- VGG (very deep CNN, 16-19 weight layers), 2015
- GoogLeNet (22 layers), 2015
- Deep Residual Net (152 layers), 2015

<https://blogs.nvidia.com/blog/2016/01/12/accelerating-ai-artificial-intelligence-gpus/>

# CNN learns hierarchical representations



H. Lee et al. (2009), "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," ICML.

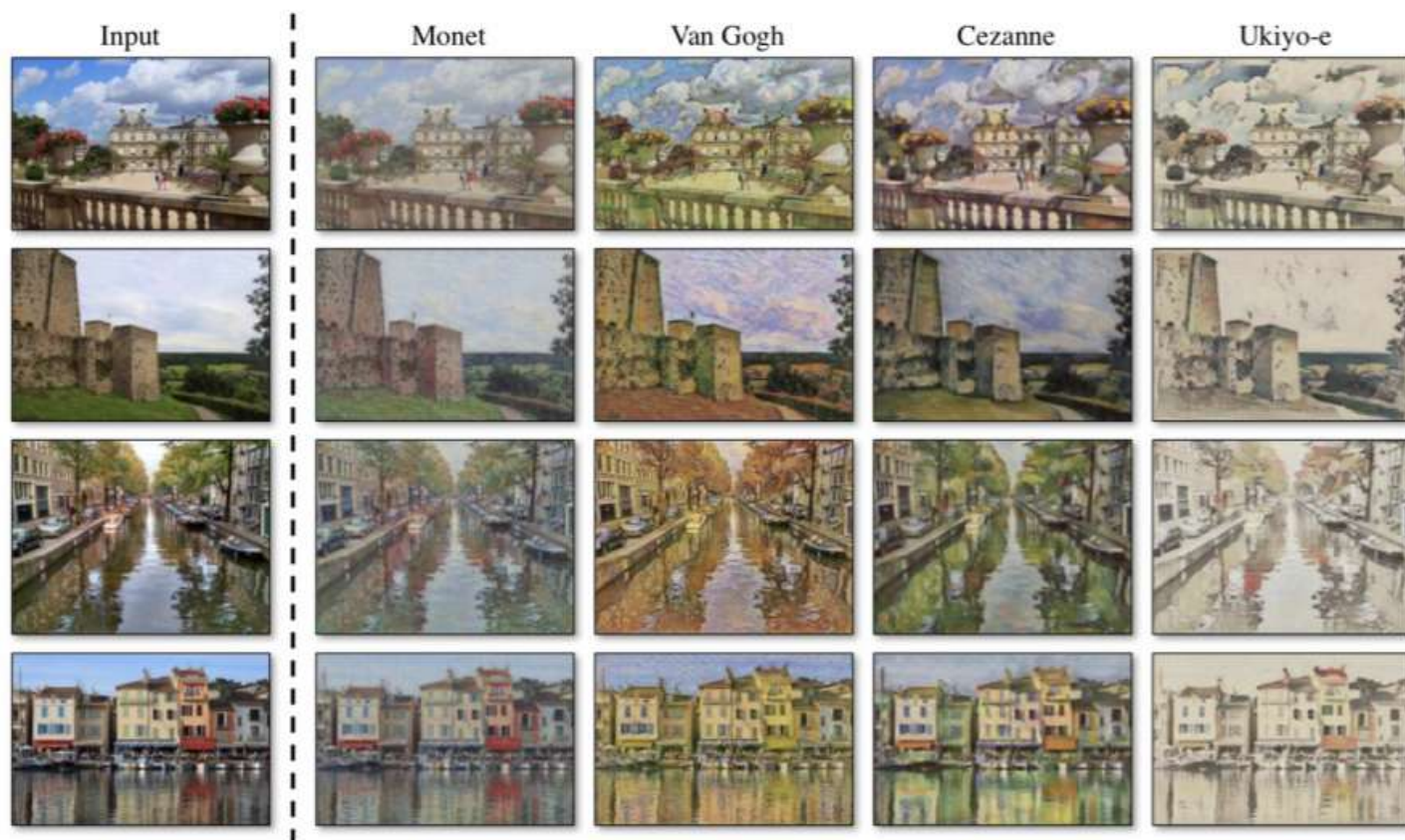
# Deep reinforcement learning

Playing Atari [Google DeepMind, 2013]:



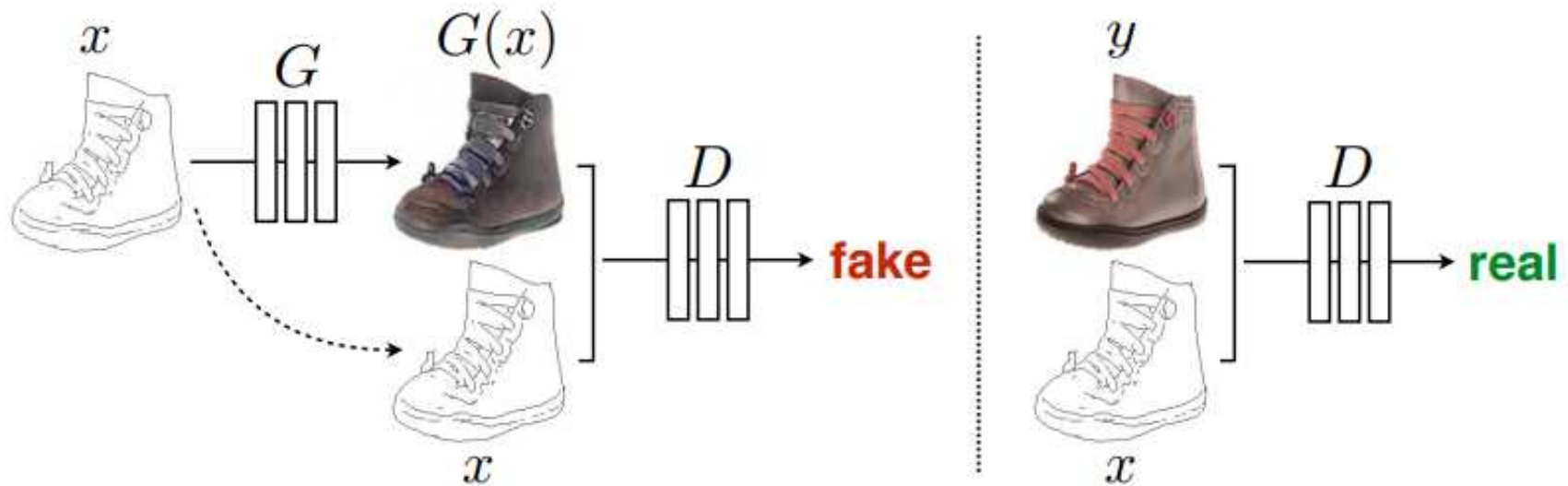
- Just use a neural network for  $\hat{Q}_{\text{opt}}(s, a)$
- Last 4 frames (images)  $\Rightarrow$  3-layer NN  $\Rightarrow$  keystroke
- $\epsilon$ -greedy, train over 10M frames with 1M replay memory
- <https://www.youtube.com/watch?v=V1eYniJ0Rnk>

# Style transfer

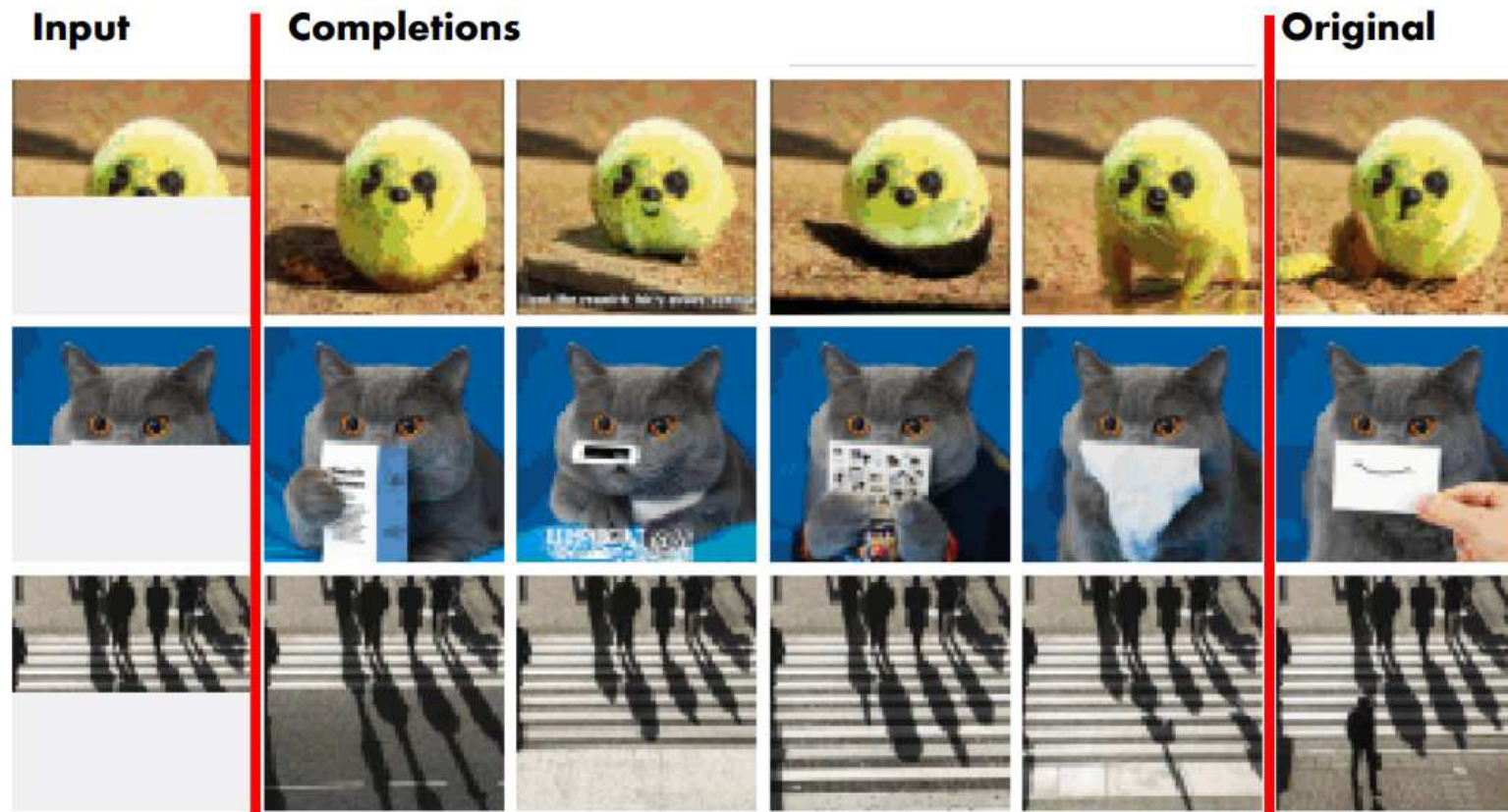




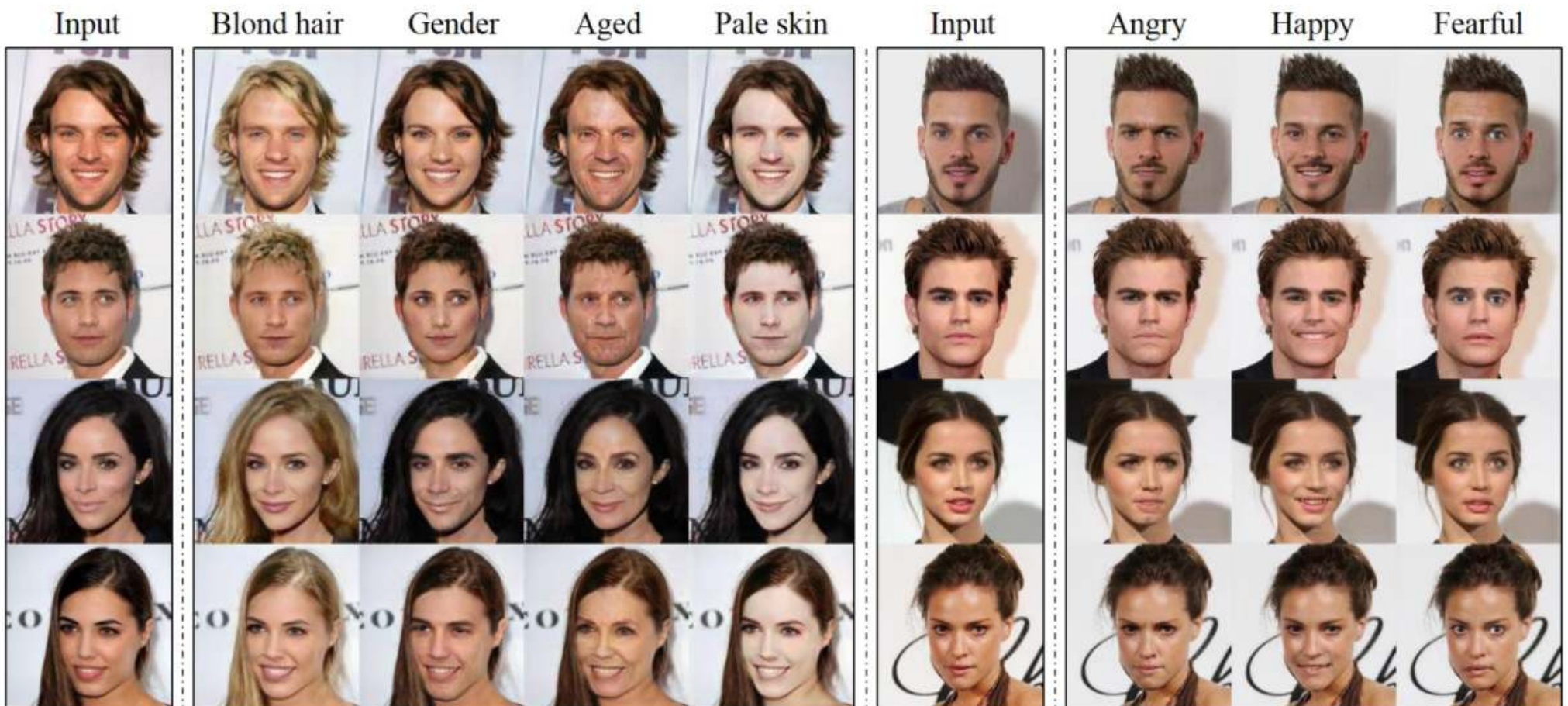
## Complete drawing from sketch



## Image completion (Image-GPT, 2020)



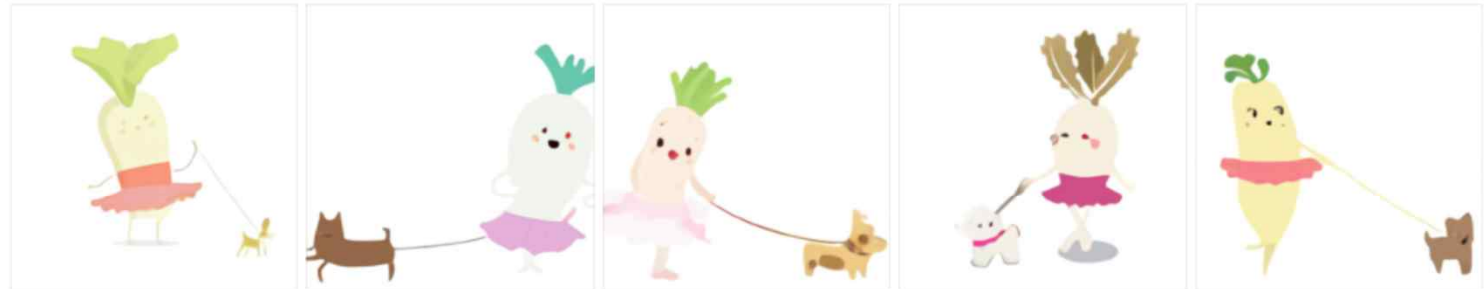
# Image manipulation



# Drawing from text description (Dall-E, 2021)

TEXT PROMPT an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED  
IMAGES



TEXT PROMPT an armchair in the shape of an avocado. . . .

AI-GENERATED  
IMAGES



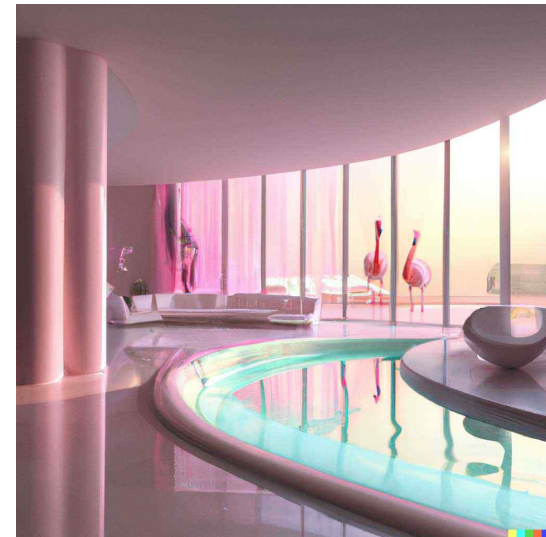
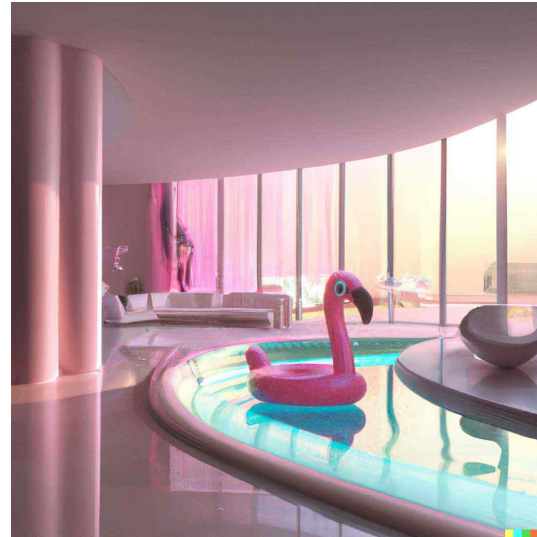
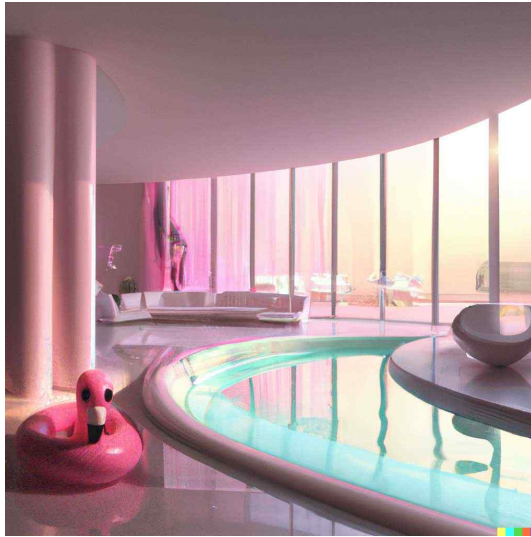


## Drawing from text description (Dall-E2, 2022)

An astronaut riding a horse in  
a photorealistic style



Select location to add flamingo



## GPT-3 (2020)

- If you wanted to make a bunch of money as a result of all of this, what would you do?
- The next best thing to do is to own a business that requires little human interaction. For example, an online store that sells books, clothes or electronic goods would be perfect.
- What companies do you think would do very well during this time of pandemic?
- There are several. Tesla, Google and Netflix are all likely to do well. They are doing things that require little human interaction. Google is working on self driving cars, these will be a part of the solution in the future. Self driving cars will become popular because the automation will reduce congestion and also will limit human-to-human contact.

## GPT-3 (2020)

- How many eyes does a giraffe have?
  - A giraffe has two eyes.
- How many eyes does my foot have?
  - Your foot has two eyes.
- How many eyes does a spider have?
  - A spider has eight eyes.
- How many eyes does the sun have?
  - The sun has one eye.

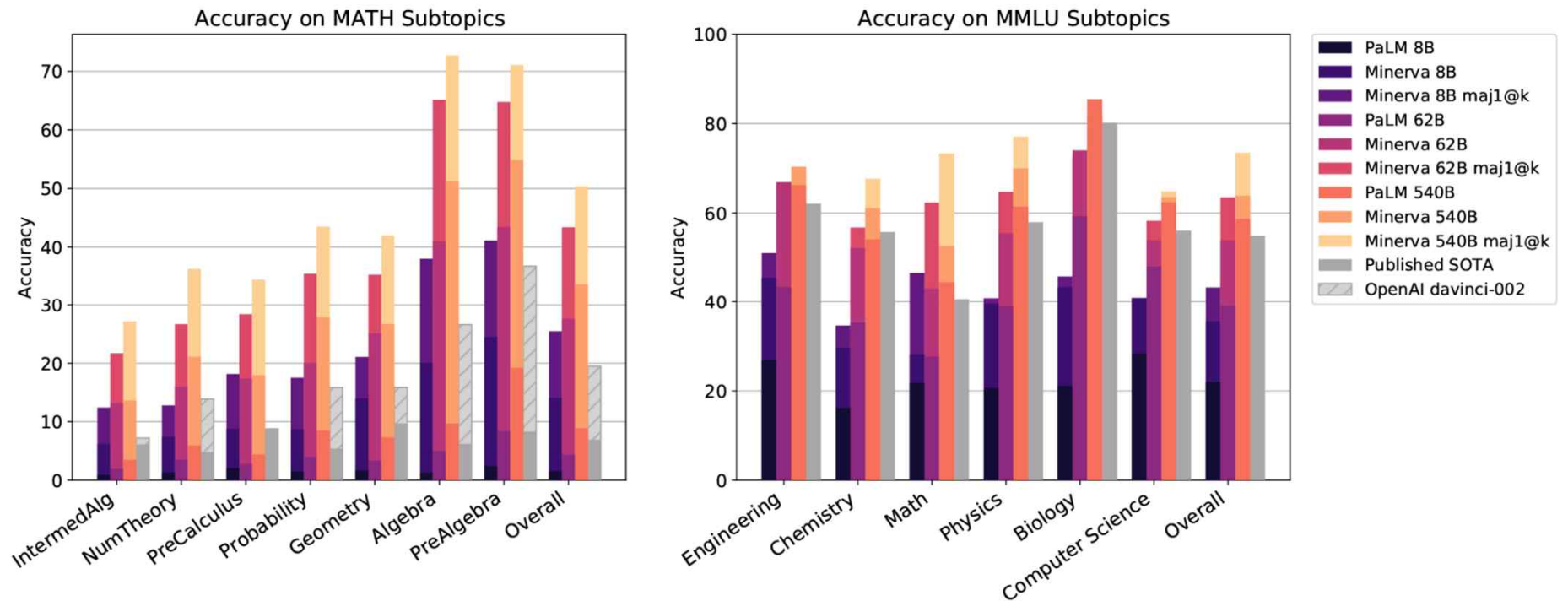
# Deep learning limitations ?

Books, articles, videos were released at 2019 saying that

- Deep learning has limitations!
- Learning-based methods cannot do commonsense reasoning.
- “By 2020, the popular press starts having stories that the era of Deep Learning is over”



## Minerva, July 2022 by Google Research



80% accuracy on 10-digit addition, and 20% accuracy on 18-digit addition

# ChatGPT (2022)



Can you write me a 4 paragraph academic essay comparing and contrasting the theories of nationalism of Benedict Anderson and Ernest Gellner?



Benedict Anderson and Ernest Gellner both have theories about the origins and nature of nationalism, but their perspectives differ in key ways. Anderson believes that nationalism is a cultural construct, arising from the development of print capitalism and the creation of imagined communities. Gellner, on the other hand, sees nationalism as a product of modernization and industrialization, with nations being formed as a means of organizing society and providing a shared identity.

One key difference between the two theories is the role of culture. For Anderson, culture is central to the creation of nationalism, as the spread of print media allows for the creation of imagined communities that transcend physical boundaries. This allows for the development of a national identity based on shared cultural experiences and practices. In contrast, Gellner downplays the importance of culture in the formation of nationalism, arguing that it is primarily a function of economic and political developments. He sees nations as being formed in response to the need for a shared identity and sense of belonging in an industrialized society.

Another point of contrast is the emphasis on agency. Anderson's theory suggests that people actively participate in the construction of national identity

# ChatGPT (2022)

- Based on data up to 2021.
- No answer like “How is the weather?”
- Weak at logical reasoning like easy math questions
- “Hallucinates” facts
- Failure cases:  
<https://github.com/giuven95/chatgpt-failures>

H 대한민국에서 정보검색 전문가 추천해줘

GPT 대한민국에서 정보검색 전문가를 추천해드리겠습니다.

1. 김성훈 교수 (서울대학교 컴퓨터공학부) **실존하지 않음**

- 자연어 처리, 정보검색, 강화학습 등 분야에서 활약하는 교수로 유명합니다.
- 대한민국에서는 딥러닝을 이용한 자연어 처리 분야에서 권위적인 지위를 가지고 있습니다.

2. 강민국 교수 (한양대학교 정보컴퓨터공학과)

- 정보검색, 데이터 마이닝, 자연어 처리 등의 분야에서 활동하고 있습니다.
- 현재 한양대학교 데이터 사이언스 연구소장으로도 활동하고 있습니다.

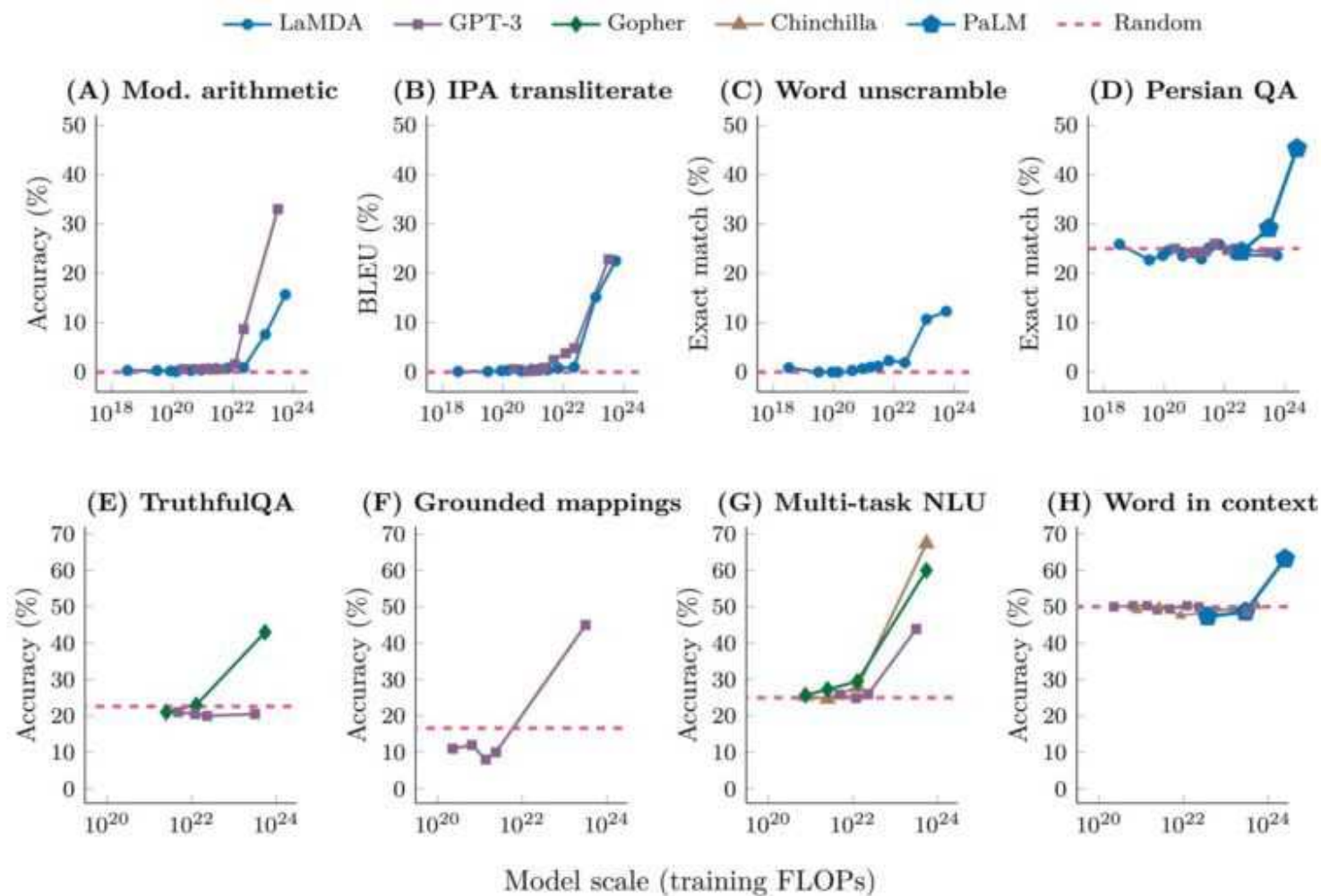
G I married my mother's daughter in law. How is that possible?

GPT It is not possible for a person to marry their mother's daughter-in-law. A daughter-in-law is the wife of a person's son, so marrying a daughter-in-law would be equivalent to marrying one's own child, which is not allowed.

## GPT3.5 (ChatGPT) -> GPT4

- Handles longer inputs (token: 2,000 -> 8,000(standard) or 32,000(extended))
- Handles image input and output (Dall-E2 기능)
- Solves harder questions (LSAT 10% -> 90%, Bio-Olympiad 31% -> 99%)
- Better and more creative (writing, tutoring, coding, etc)

# Emergent ability in LLMs



## Deep learning milestones:

- 1950-: Perceptron, Neural network
- 1970-: Backpropagation, RBM, RNN
- 1980-: CNN, LSTM
- 2006: “Deep learning”, DBN
- 2009: ImageNet
- 2012-15 (CV): AlexNet, GoogLeNet, ResNet, GAN
- 2016~17: AlphaGo, AlphaZero
- 2017-20 (NLP): Transformers, BERT, GPT3
- 2021-22: Diffusion model, Visual language models
- 2023: ChatGPT, GPT4, sLLMs
- 2024: Gemini, Sora, AlphaGeometry, ..

## Milestones:

- Earth created 4.54 billion years ago
- Modern humans 300,000 years ago
- Industrial revolution 200 years ago

# Transformer-based language models

- BERT (Google, 2017) 이후
- GPT-3 (OpenAI, 2020) 이후
- ChatGPT (GPT3.5, 2023) 이후
- XLNet (Google/CMU)
- Gopher (Google DeepMind)
- GPT4
- RoBERTa (Facebook)
- OPT-175B (Meta, same size as GPT-3 but open to public)
- Bard
- DistilBERT (HuggingFace)
- MT-NLG (Microsoft & NVIDIA)
- Alpaca
- CTRL (Salesforce)
- LaMDA2 (Google, targeted for improving Google Search)
- Vicuna
- **GPT-2 (OpenAI)**
- 하이퍼클로바 (네이버)
- GPT4ALL
- ALBERT (Google)
- KoGPT (카카오)
- ... and ...
- Megatron (NVIDIA)
- EXAONE (LG, 전문용)
- lots of applications..
- **“AI Cambrian period”**

# Limitations of LLM and generative AI

1. Easy to create data (text, images, videos) => data quantity explosion but quality deterioration => filtering good data costs (e.g. examination cost of novel manuscript, cost of filtering out fake news).
2. Original data copyright issue => original data market fall
3. Environmental issue: training GPT3 => 285,000CPU, 10,000 GPU, 400GB/s connectivity => 550 tons CO2 generation, 120-year electricity consumption of a household
4. Heavy, high resource cost
5. Hard to avoid hallucinations in Transformer-based LLMs
6. Outdated information problem
7. Security, authority control issue => Companies/organizations need to develop their own AI models

\*Approaches like Retrieval Augmented Generation (RAG) is researched to relieve the problems of 4,5,6,7.



# AI Applications

...

Web search  
Speech recognition  
Handwriting recognition  
Machine translation  
Information extraction  
Document summarization  
Question answering  
Spelling correction  
Image recognition  
3D scene reconstruction  
Human activity recognition  
Autonomous driving  
Music information retrieval  
Automatic composition  
Social network analysis

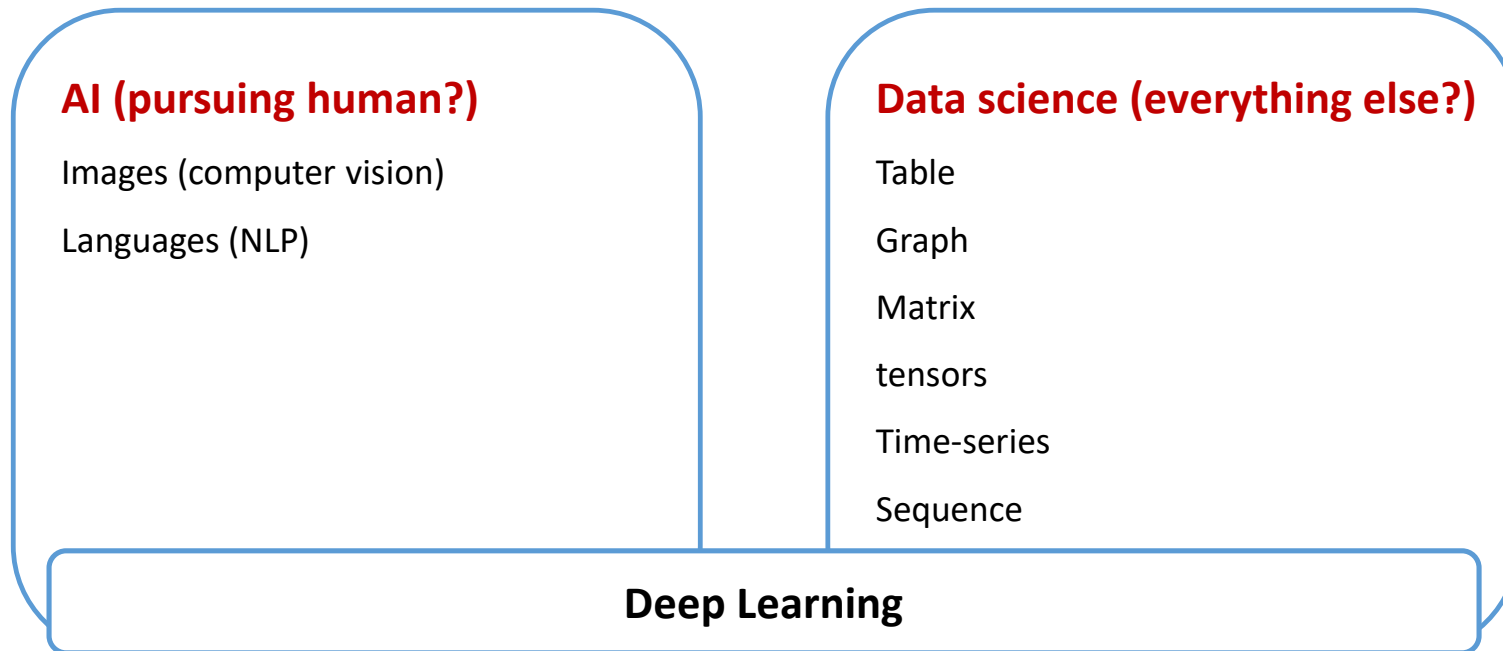
...

...

Product recommendation  
Advertisement placement  
Smart-grid energy optimization  
Household robotics  
Robotic surgery  
Robot exploration  
Spam filtering  
Fraud detection  
Fault diagnostics  
AI for video games  
Financial trading  
Dynamic pricing  
Protein folding  
Medical diagnosis  
Medical imaging

...

# AI vs data science



## Major conferences for AI & data science

- AI: AAAI, IJCAI
- Machine learning: NeurIPS (NIPS), ICLR, ICML
- Computer vision: CVPR, ICCV, ECCV
- Natural language processing: ACL, EMNLP, NAACL
- Data science: KDD, ICDM, WWW
- Information retrieval: SIGIR, CIKM

*How to implement artificial intelligence?*

# Two sources of AI challenges

## **Computational challenge**

- Most AI problems are NP-hard
- Go –  $361^{200}$  trajectories that a player would have to consider to play optimally.

## **Information challenge**

- Translate a sentence
- Classify a bird from image

# Resources we need

Computation (time/memory)



Information (data)



# Software development

Real-world task

Algorithm



```
# Data structure for supporting uniform cost search.
class PriorityQueue:
    def __init__(self):
        self DONE = -100000
        self heap = []
        self priority = {} # Map from state to priority

    # Insert (state) into the heap with priority (newPriority). If
    # (state) isn't in the heap or (newPriority) is smaller than the existing
    # priority.
    # Returns whether the priority queue was updated.
    def update(self, state, newPriority):
        oldPriority = self.priority.get(state)
        if oldPriority == None or newPriority < oldPriority:
            self.priority[state] = newPriority
            heapq.heappush(self.heap, (newPriority, state))
            return True
        return False

    # Returns (state) with minimum priority, (priority)
    # or (None, None) if the priority queue is empty.
    def popmin(self):
        while len(self.heap) > 0:
            priority, state = heapq.heappop(self.heap)
            if self.priority.get(state) == self.DONE: continue # outdated priority, skip
            self.priority[state] = self.DONE
            return (state, priority)
        return (None, None) # Nothing left...

=====
# Simple examples of search problems to test your code for Problem 2.

# A simple search problem on the number line:
# Start at 0, want to go to 10, cost:1 to move down, 1 to move up.
class NumberLineSearchProblem:
    def startState(self): return 0
    def isGoal(self, state): return state == 10
    def successors(self, state): return [(state-1, 1), (state+1, 1)]

# Function to create search problems from a graph.
# You can use this to test your algorithm.
def createSearchProblemFromGraph(start, goal, description):
    # Parse the graph.
    lines = description.splitlines()
    for line in description.splitlines():
        if line[0] == '#': continue # skip comments
        # Line from state s to state t.
        s, d, cost = line.split()
        cost = float(cost)
        # Add to the same as the destination state (t).
        graph[s].append((t, cost))

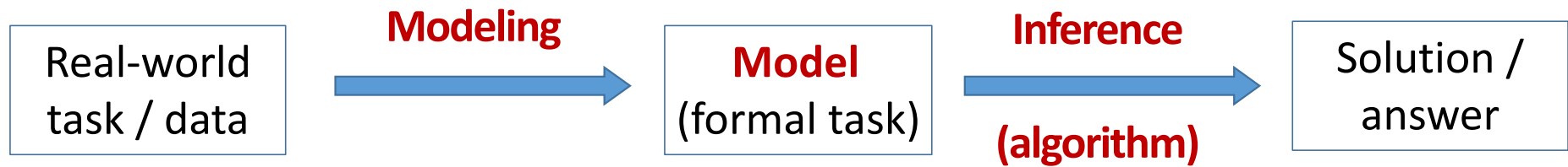
    # Create the search problem.
    return SearchProblem(graph, start, goal)
```

# Algorithms (example)

- Formal task:
  - Input: list  $L = \{x_1, \dots, x_n\}$  and a function  $f : X \rightarrow \mathbb{R}$
  - Output:  $k$  highest-scoring elements
- Example ( $k = 3$ ):
  - $L : \quad A \quad B \quad C \quad D$
  - $f : \quad 3 \quad 2 \quad 7 \quad 1$
- Two algorithms:
  1. Scan through to find the largest, scan through again to find the second largest, etc.
  2. Sort  $L$  based on  $f$ , return first  $k$  elements



# AI development

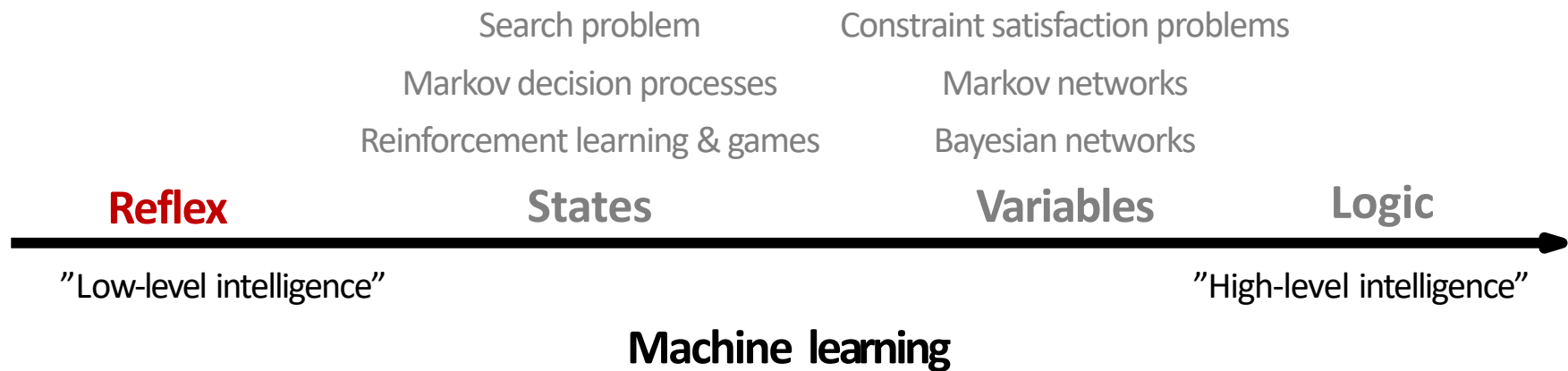


# Modeling (example)

- Real-world task:
  - Input: 20 billion web pages, a keyword query
  - Output: 10 most relevant web pages
- Modeling:
  - $L$  = list of web pages
  - $f(x) = 10 * \text{QueryMatch}(x) + 3 * \text{PageRank}(x)$
- Formal task:
  - Input: list  $L = \{x_1, \dots, x_n\}$  and a function  $f : X \rightarrow \mathbb{R}$
  - Output:  $k$  highest-scoring elements

# Modeling and algorithms

- Separate **what** to compute (**modeling**) from **how** to compute it (**algorithms**) => Advantage: division of labor
- What do we learn?
  - Type of models...
  - Art of modeling...
  - Developing Algorithms...



## Reflexed-based models

- Examples: linear classifiers, deep neural networks
- Application example: sentiment analysis

- Input: movie review

*“Shows moments of promise but ultimately succumbs to cliches and pat storytelling.”*

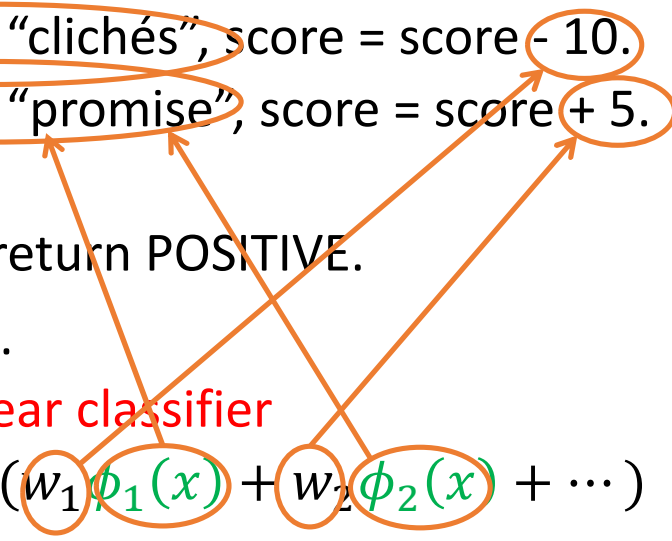
- Output: sentiment

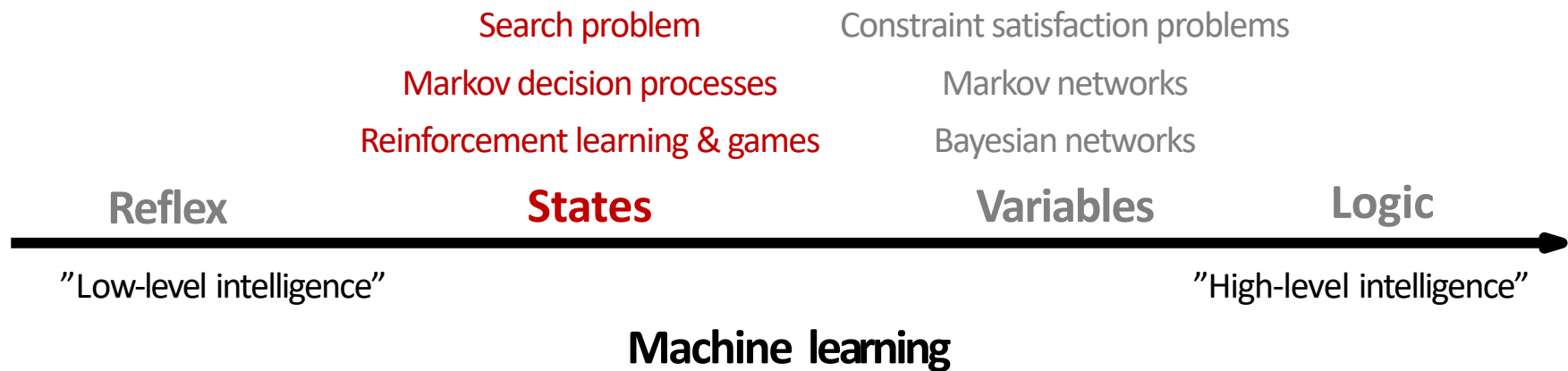
*POSITIVE or NEGATIVE*

## Sentiment analysis model example

- Input:  $x$ , a document or sentences
- Output:  $f(x)$ , a simple function of  $x$
- Example: **model  $f$  is a set of simple rules**
  - If  $x$  contains “clichés”, return NEGATIVE.
  - If  $x$  contains “promise”, return POSITIVE.
  - ...

## Sentiment analysis model example: linear model

- Let's take every word into account...
  - Output  $f$  is determined based on scores
    - Set score = 0
    - If  $x$  contains "clichés", score = score - 10.
    - If  $x$  contains "promise", score = score + 5.
    - ...
    - If score > 0, return POSITIVE.
  - More generally...
    - Key idea: linear classifier
    - $f(x) = \text{sign}(w_1\phi_1(x) + w_2\phi_2(x) + \dots)$
  - How about "not bad"?
- 





# Text Reconstruction

- Chinese is written without spaces:

是 什 4 意思

- Arabic omits (some) vowels:

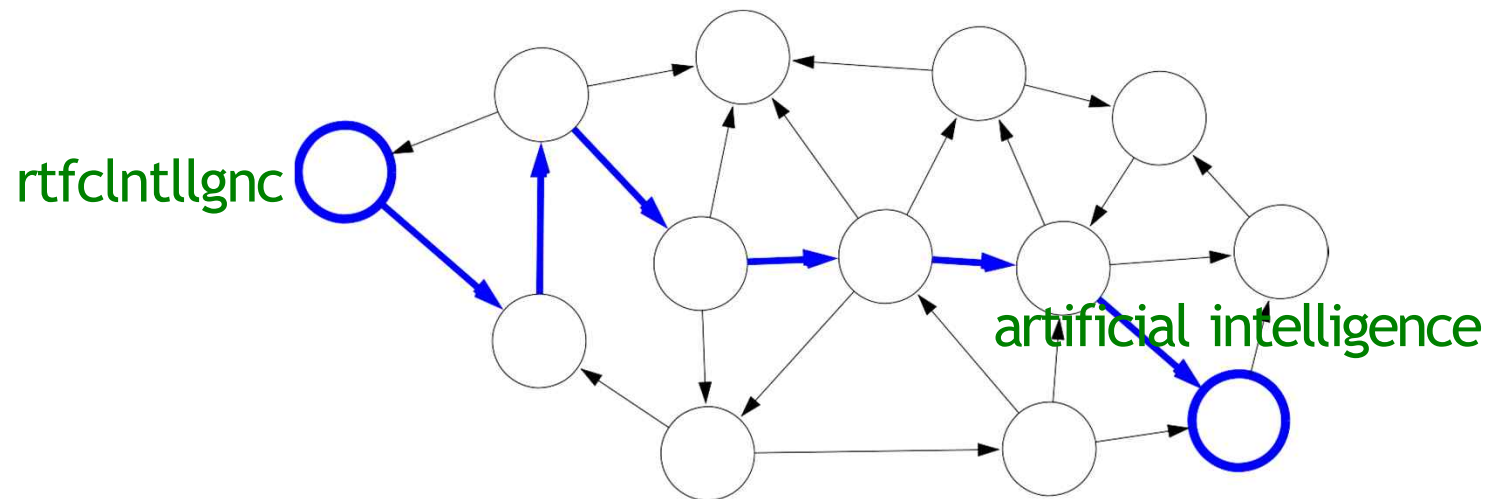
مَكْتَبَة

- Remove vowels and spaces from an English phrase:

rtfclntllgnc

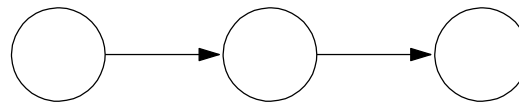
# State-based Models

- Solutions are represented as paths through a graph

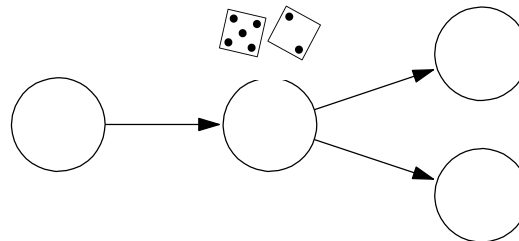


# State-based Models

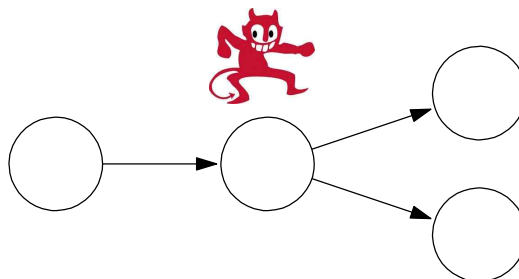
- Search problems: you control everything



- Markov decision processes: against nature (e.g., Blackjack)



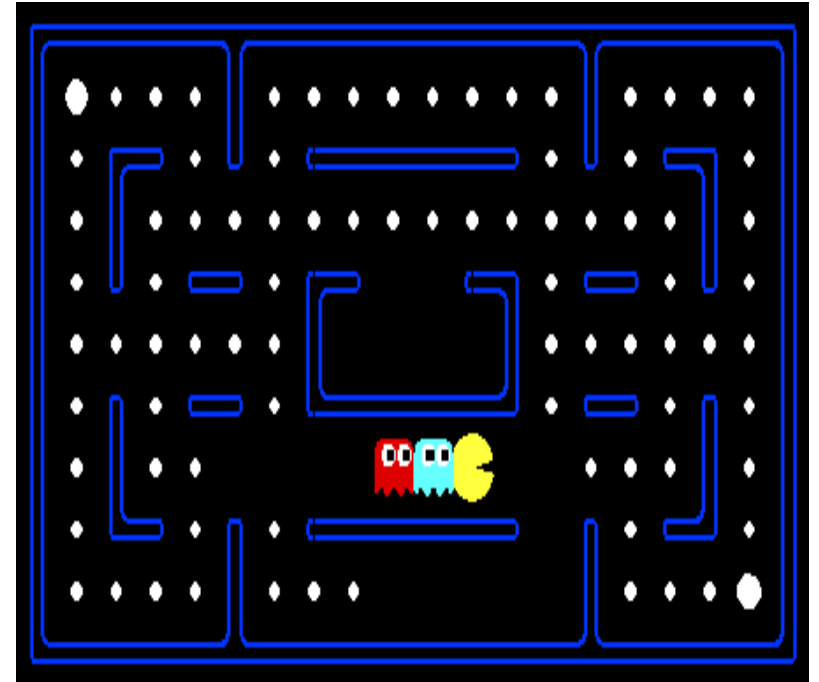
- Adversarial games: against opponent (e.g., chess)



# Pac-Man

What kind of model is appropriate for playing Pac-Man against ghosts that move into each valid adjacent square with equal probability?

1. Search problem
2. Markov decision process
3. Adversarial game

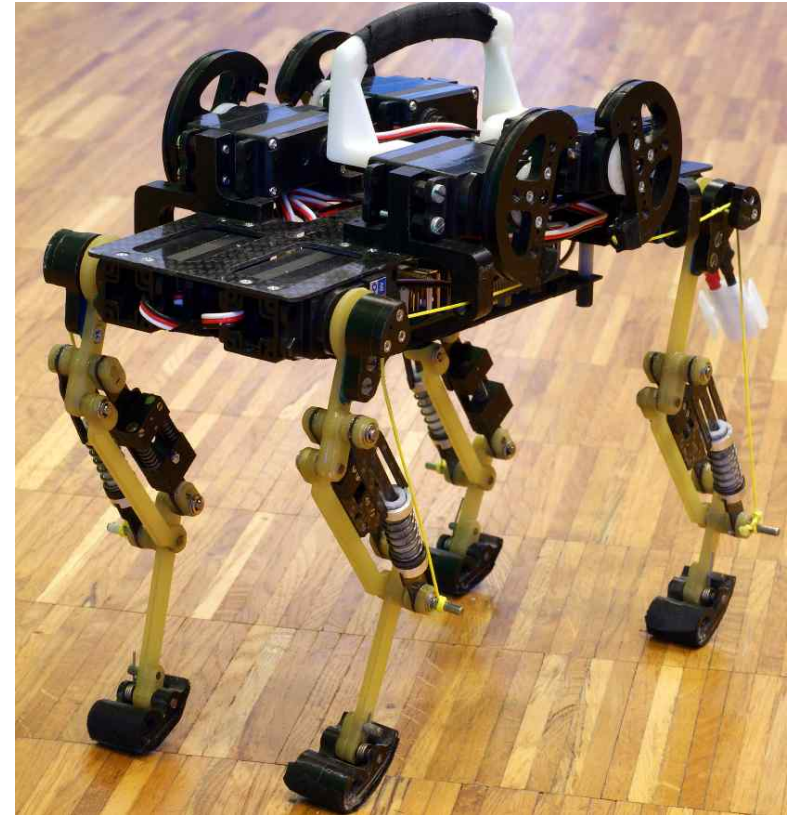


# Crawling robot

Goal: maximize distance travelled by robot

Markov decision process (MDP):

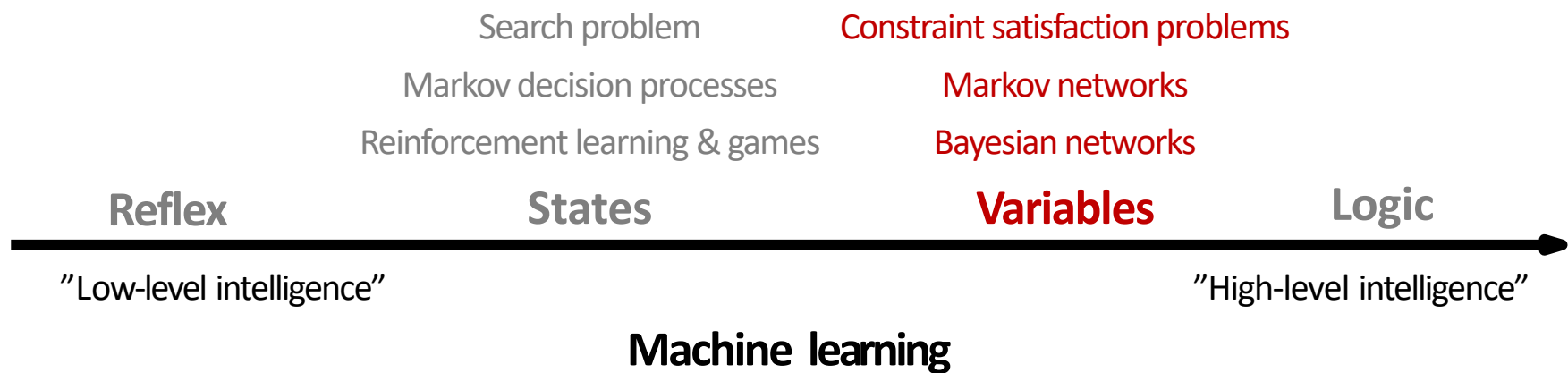
- States: positions (4 possibilities) for each of 2 servos
- Actions: choose a servo, move it up/down
- Transitions: move into new position (unknown)
- Rewards: distance travelled (unknown)



# AlphaGo



- Supervised learning: on human games
- Reinforcement learning: on self-play games
- Evaluation function: convolutional neural network (value network)
- Policy: convolutional neural network (policy network)
- Monte Carlo Tree Search: search / lookahead



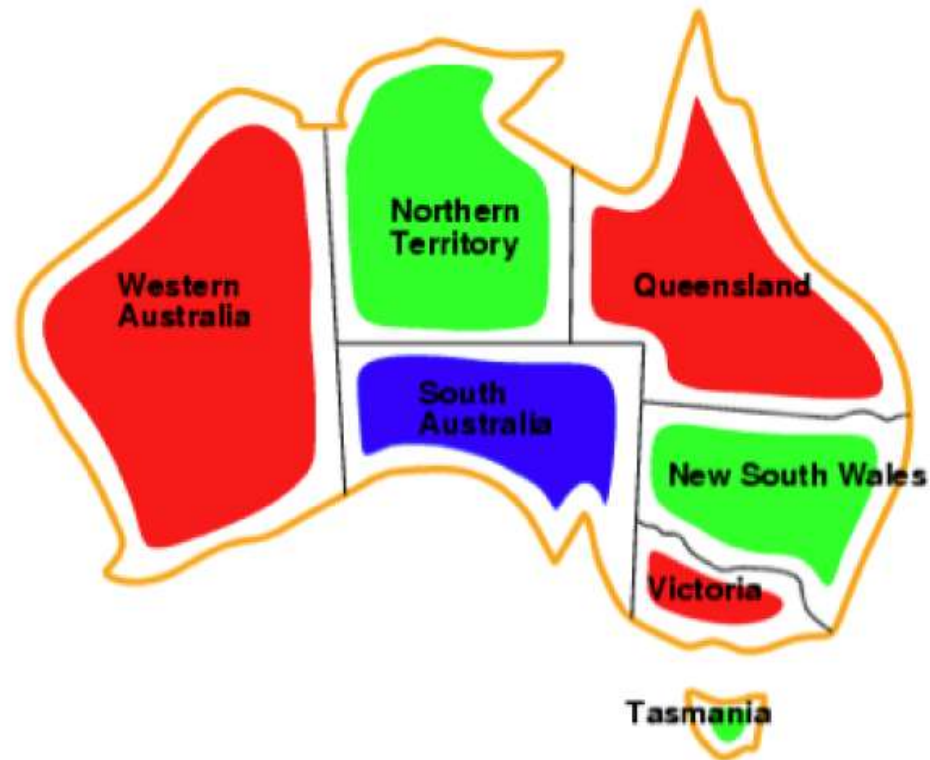
# Constraint Satisfaction Problem (CSP)



**Question:** how can we color each of the 7 provinces {red, green, blue} so that no two neighboring provinces have the same color?



# Map coloring



(one possible solution)

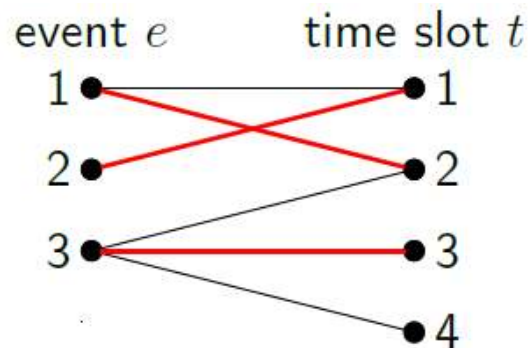
# Constraint Satisfaction Problem (CSP)

Three sculptures (A, B, C) are to be exhibited in rooms 1, 2 of an art gallery.

The exhibition must satisfy the following conditions:

- Sculptures A and B cannot be in the same room.
- Sculptures B and C must be in the same room.
- Room 2 can only hold one sculpture.

# Event scheduling

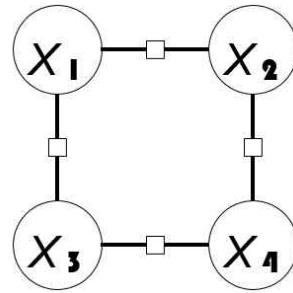


Setup:

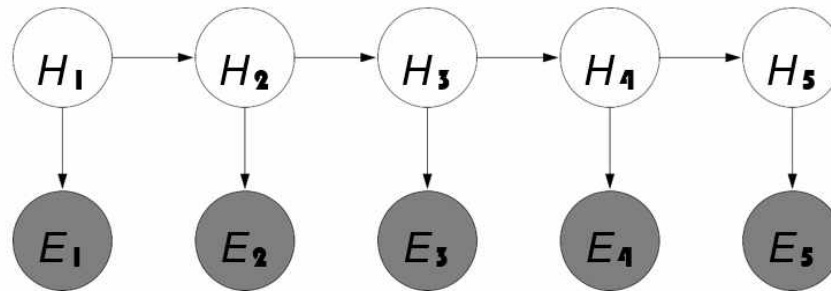
- Have  $E$  events and  $T$  time slots
- Each event  $e$  must be put in **exactly one** time slot
- Each time slot  $t$  can have **at most one** event
- Event  $e$  allowed in time slot  $t$  only if  $(e, t) \in A$

# Variable-based Models

- **Constraint satisfaction problem**: hard constraints (e.g., map coloring, scheduling)



- **Bayesian networks**: soft dependencies (e.g., tracking cars from sensors)



# Topic modeling

**Question:** given a text document, what topics is it about?

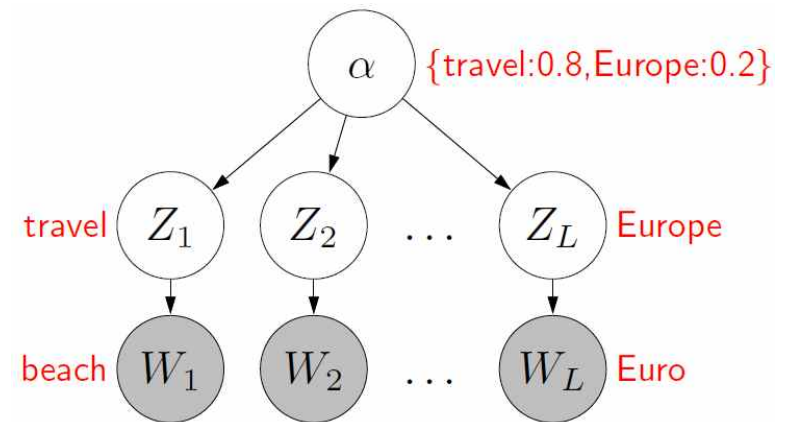
**Probabilistic program:** latent Dirichlet allocation

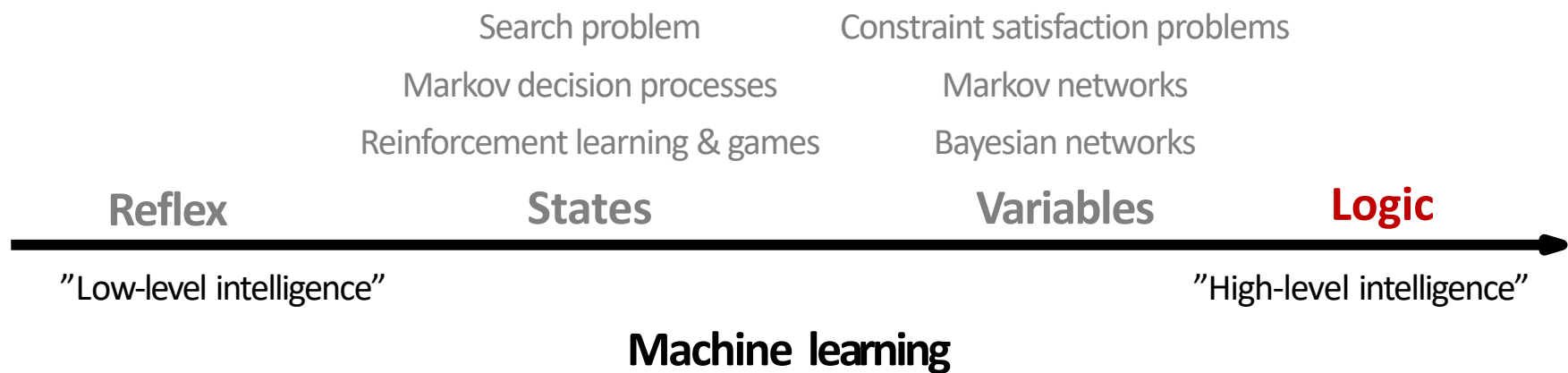
Generate a distribution over topics  $\alpha \in \mathbb{R}^K$

For each position  $i = 1, \dots, L$ :

Generate a topic  $Z_i \sim p(Z_i|\alpha)$

Generate word  $W_i \sim p(W_i|Z_i)$





# Question

You get extra credit if you write a paper and you solve the problems.

You didn't get extra credit, but you did solve the problems.

Did you write a paper?

Yes or No

# Knowledge representation and reasoning

All students work hard.

John is a student.

Therefore, John works hard.

Variable-based models would explicitly represent all the students – this is inefficient

Need **expressive power** of logic to represent this ...



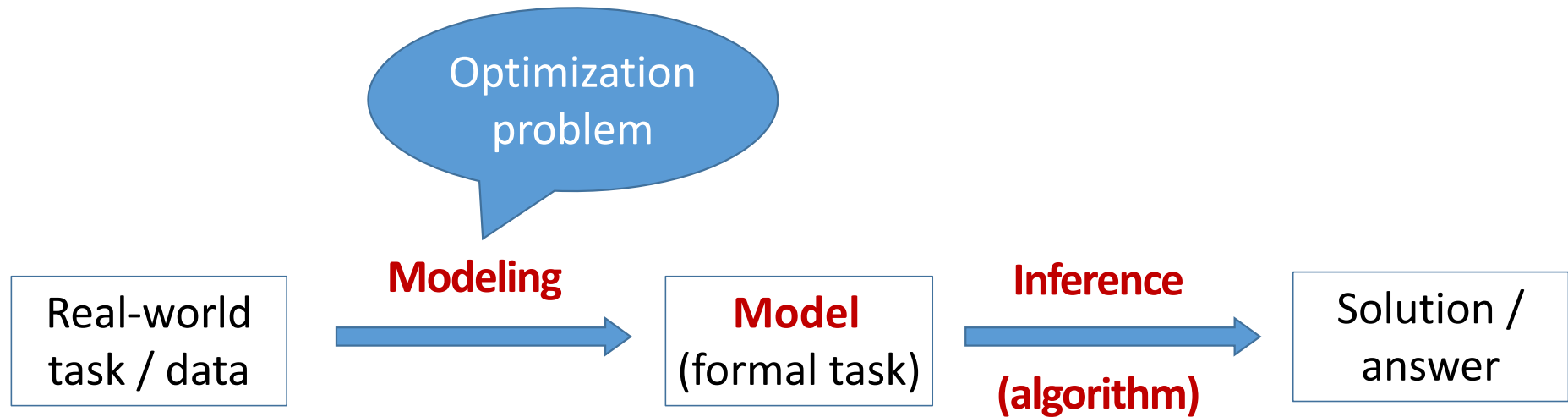
# Logic-based model: smart personal assistant



Need to:

- Digest information and build knowledge (**knowledge representation**)
- Reason deeply with the knowledge and make inferences (**reasoning**)

# AI development



## Problem: predicting exam score $y$

- $x$ : # of hours studying
- $y$ : exam score
- Assume  $y = wx$
- $w$  is a **learning parameter** we need to estimate (learn) from training data
- $D_{\text{train}}$ : set of pairs  $\{(x_1, y_1), \dots, (x_n, y_n)\}$
- $f$ :  $w \in \mathbb{R}$  that minimizes the squared error  $F(w) = \sum_{i=1}^n (x_i w - y_i)^2$
- Example:

$$\{(2,4)\} \Rightarrow 2$$

$$\{(2,4),(4,2)\} \Rightarrow ?$$

# Optimization

- Optimization problem:

$$\min_{x \in C} F(x)$$

- Discrete optimization:  $x$  is a discrete object

$$\min_{x \in \{abcd, xyz\}} \text{Length}(x)$$

Algorithmic tool: dynamic programming

- Continuous optimization:  $x$  is a vector of real numbers

$$\min_{x \in \mathbb{R}} (x - 5)^2$$

Algorithmic tool: gradient descent

# Summary

- Applications of AI: high-impact, diverse
- Challenges: computational/information challenges
- Paradigm: modeling + inference (algorithm)
- Models: [reflex, states, variables, logic] + learning (optimization)