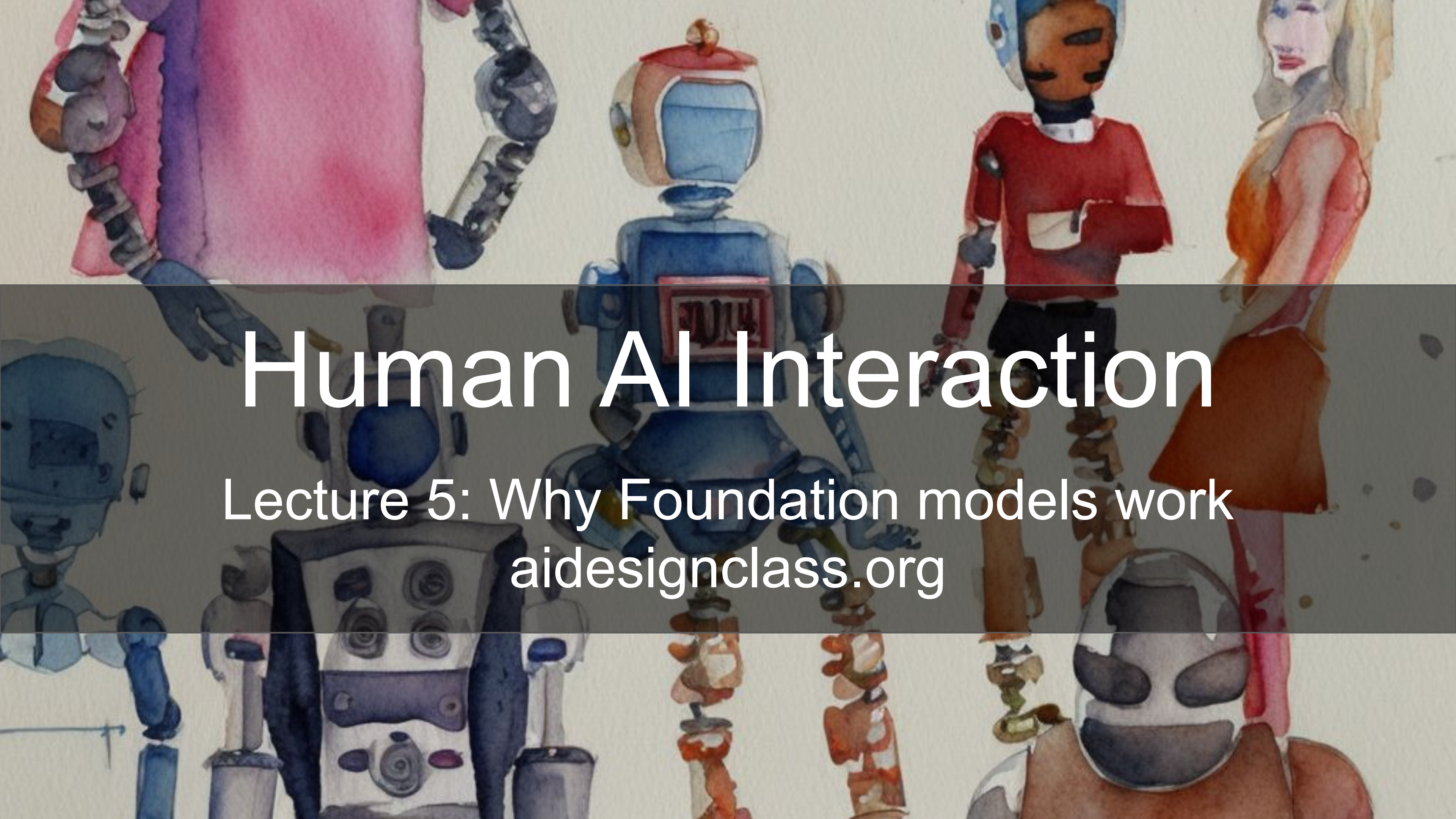


Human AI Interaction

Lecture 5: Why Foundation models work

aidesignclass.org

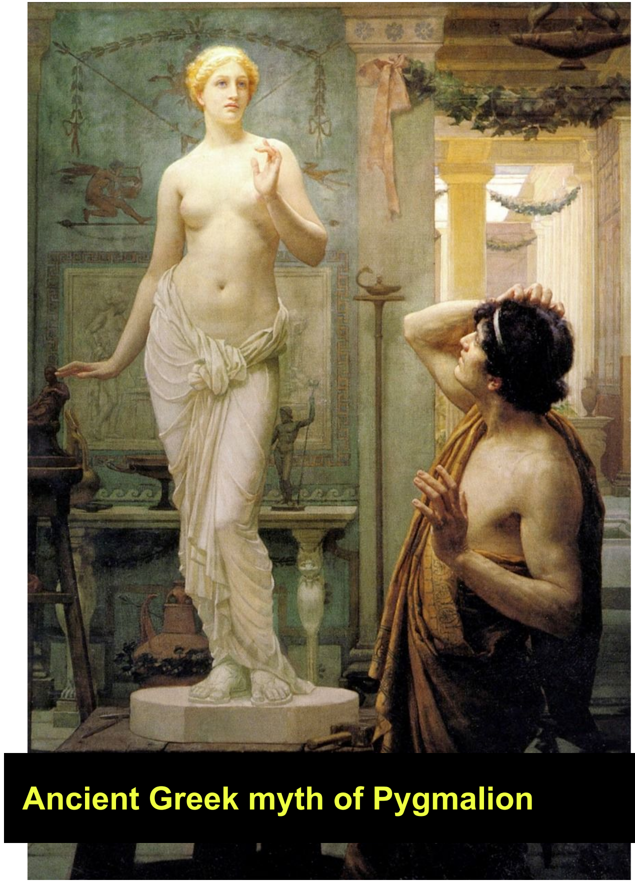
Recap + Plan

- Journey maps: A start-from-the-user method
- Tech matching: a start-from-tech method
- Understanding how LLMs work – predicting the next word can be powerful

Today:

- ~45 min: History of AI/HCI + How foundation models work
- ~30 min: Project questions + prep time

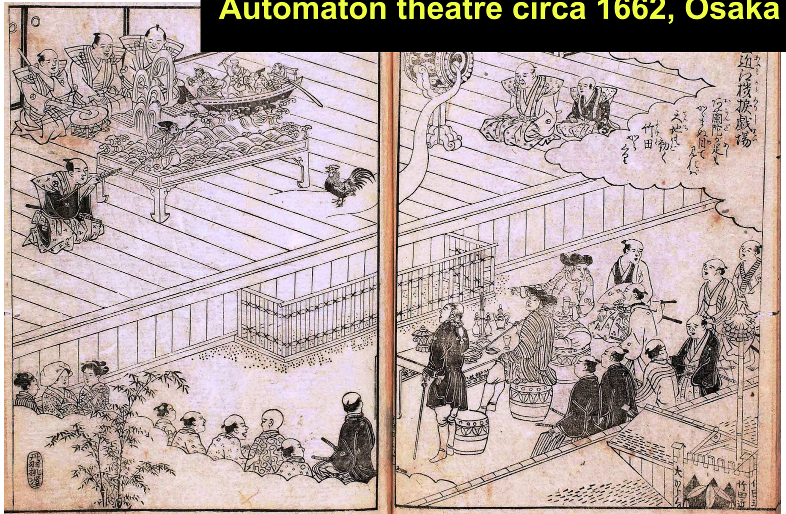
Ancient automaton



Ancient Greek myth of Pygmalion

"... by discovering the true nature of the gods, man has been able to reproduce it." - maybe some guy named Hermes Trismegistus < 200 BC

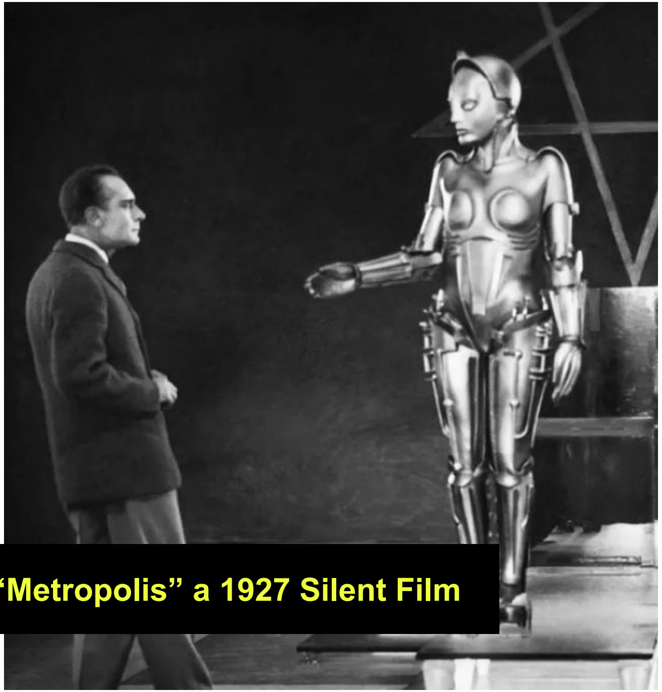
Automaton theatre circa 1662, Osaka Japan



Dreams of Robots 1860s - 1940s

“R.U.R” a 1920 play

"...the time will come when the machines will hold the real supremacy over the world." - 1863 article by Samuel Butler



“Metropolis” a 1927 Silent Film

1942 Asimov’s Laws

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

1950 - 1956 Beginnings of Computer Science

Lots of people: **What truly *is* human intelligence?**

Alan Turing: **How can we decide when a machine has achieved human-level intelligence?**

Alan Turing

1950 - 1956 Beginnings of Computer Science

Lots of people: **What truly *is* human intelligence?**

Alan Turing: **How can we decide when a machine has achieved human-level intelligence?**

The politeness convention:

If a machine behaves as intelligently as a human being, then it is as intelligent as a human being

Alan Turing

Discuss: is this the right goal for AI?

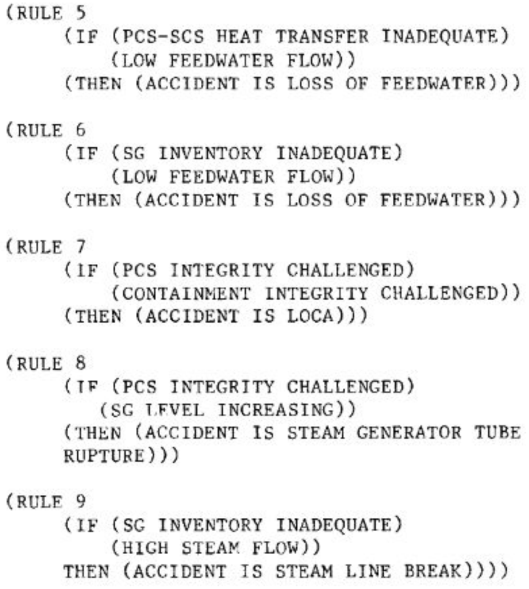
Should it be to be indistinguishable from humans?

Why?

Why not?

1956 - 1971 We can teach innate knowledge through rules

Symbolism: formal logic systems can represent intelligent action



```
(RULE 5
     (IF (PCS-SCS HEAT TRANSFER INADEQUATE)
         (LOW FEEDWATER FLOW))
     (THEN (ACCIDENT IS LOSS OF FEEDWATER)))

(RULE 6
     (IF (SG INVENTORY INADEQUATE)
         (LOW FEEDWATER FLOW))
     (THEN (ACCIDENT IS LOSS OF FEEDWATER)))

(RULE 7
     (IF (PCS INTEGRITY CHALLENGED)
         (CONTAINMENT INTEGRITY CHALLENGED))
     (THEN (ACCIDENT IS LOCA)))

(RULE 8
     (IF (PCS INTEGRITY CHALLENGED)
        (SG LEVEL INCREASING))
     (THEN (ACCIDENT IS STEAM GENERATOR TUBE
     RUPTURE)))

(RULE 9
     (IF (SG INVENTORY INADEQUATE)
         (HIGH STEAM FLOW))
     THEN (ACCIDENT IS STEAM LINE BREAK))))
```

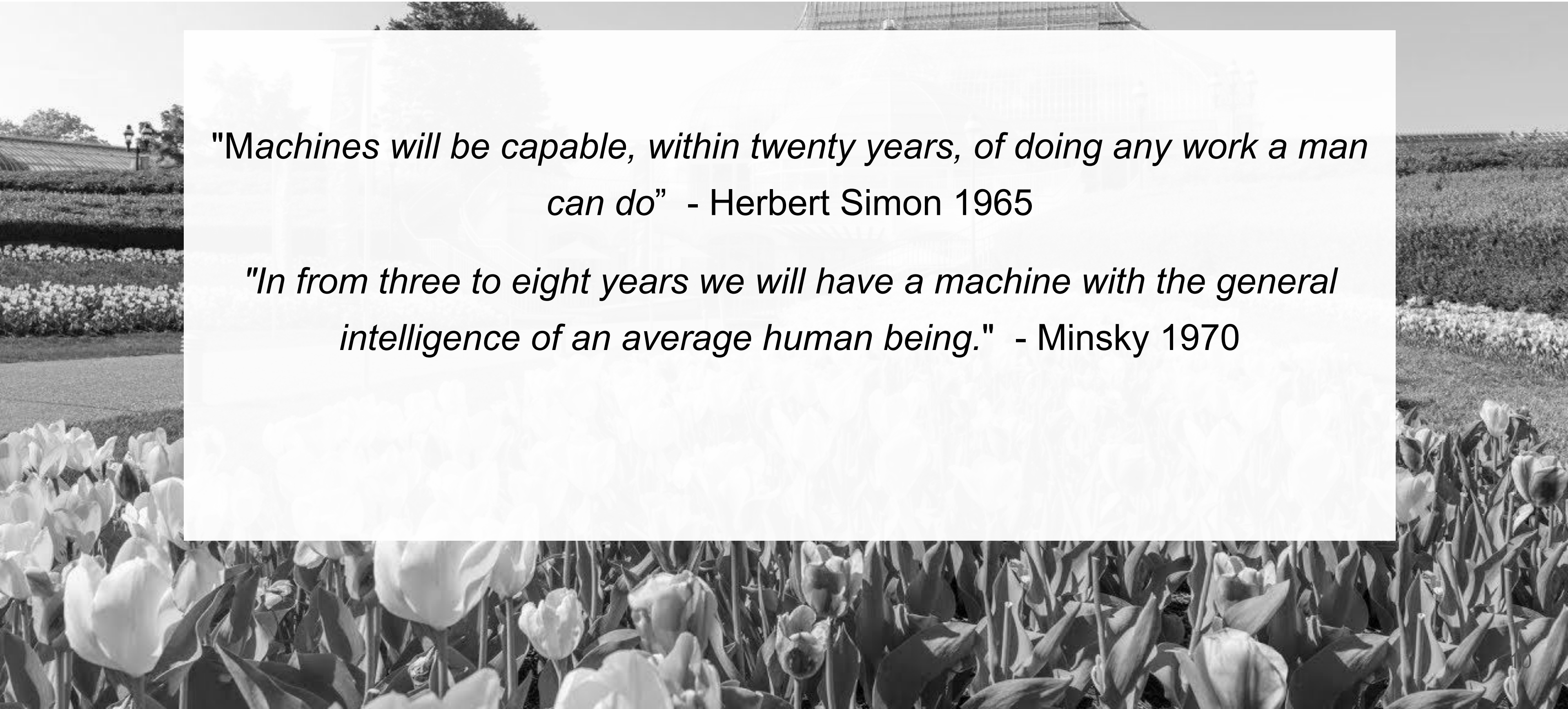
Figure 2. Event-oriented IF-THEN rules.

1956 - 1971 We can teach innate knowledge through rules

Symbolism: formal logic systems can represent intelligent action

- Newell & Simon’s “General Problem Solver” can solve math proofs by searching a logic space
- Advances in natural language processing based on rules how words relate
- Advances in computer vision based on image transforms
- Advances in robotics based on rules and search in simplified settings

PSA: Don’t make promises you can’t keep



"M*achines will be capable, within twenty years, of doing any work a man can do*" - Herbert Simon 1965

"In from three to eight years we will have a machine with the general intelligence of an average human being." - Minsky 1970

1956 - 1971 We can teach computers to learn

Connectionism: computers should mimic how the brain works

- *Neurons* make thousands of links with other neurons, making *trillions* of possible connections in the brain
- An individual neuron will fire if specific input reaches a certain threshold of electricity, otherwise no
- Threshold for a neuron to fire = *activation weights* in a neural network

Discuss: do we need to do this?

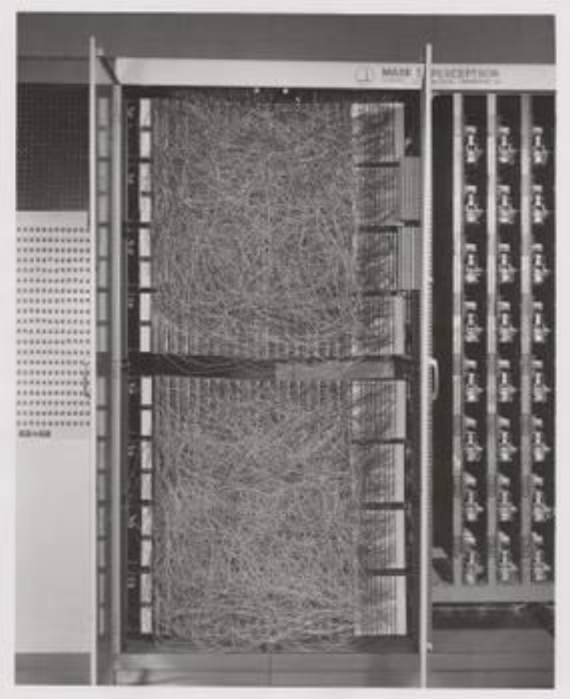
If computers must learn, should they do so as humans do?

1956 - 1971 We can teach computers to learn

The Perceptron: designed by Frank Rosenblatt 1958

”**the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.**” - NYT 1958

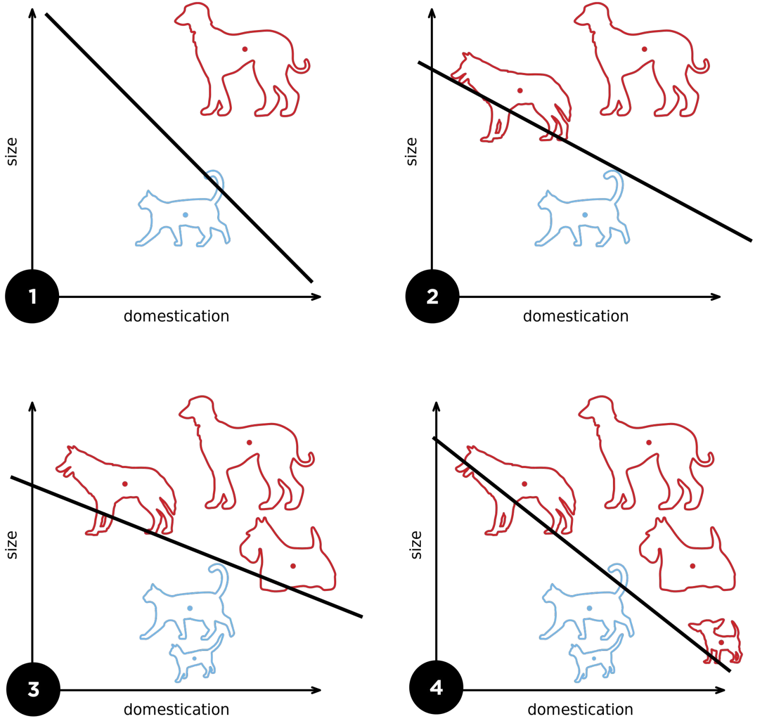
$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0, \\ 0 & \text{otherwise} \end{cases}$$



1956 - 1971 We can teach innate knowledge through rules

The Perceptron

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0, \\ 0 & \text{otherwise} \end{cases}$$



Alas - Winter arrives

Essentially none of these big ideas were quite ready for primetime.

- Lighthill Report 1973 shuts down funding in UK
- Dreyfus at MIT argues lots of human reasoning is *not based on logic rules*, involving instinct and unconscious reasoning

(Funding for AI also dried up – which is the winter that affects researchers most immediately!)

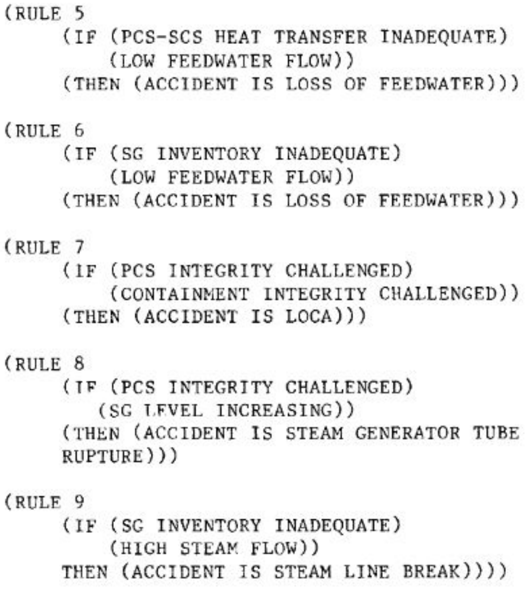
1959 Goodbye: the Perceptron “destroyed”

Minsky and Seymour Papert publish a book “Perceptrons”

- Burn piece of Perceptron approach in favor of rule approach
- Perceptron cannot handle XOR operator
- Shuts down funding for neural networks
- Rosenblatt soon dies, never to see neural nets revindicated

Formal logic can’t handle imprecision well

Symbolism: formal logic systems can represent intelligent action



```
(RULE 5
     (IF (PCS-SCS HEAT TRANSFER INADEQUATE)
         (LOW FEEDWATER FLOW))
     (THEN (ACCIDENT IS LOSS OF FEEDWATER)))

(RULE 6
     (IF (SG INVENTORY INADEQUATE)
         (LOW FEEDWATER FLOW))
     (THEN (ACCIDENT IS LOSS OF FEEDWATER)))

(RULE 7
     (IF (PCS INTEGRITY CHALLENGED)
         (CONTAINMENT INTEGRITY CHALLENGED))
     (THEN (ACCIDENT IS LOCA)))

(RULE 8
     (IF (PCS INTEGRITY CHALLENGED)
        (SG LEVEL INCREASING))
     (THEN (ACCIDENT IS STEAM GENERATOR TUBE
     RUPTURE)))

(RULE 9
     (IF (SG INVENTORY INADEQUATE)
         (HIGH STEAM FLOW))
     THEN (ACCIDENT IS STEAM LINE BREAK))))
```

Figure 2. Event-oriented IF-THEN rules.

Sussman: “*using precise language to describe essentially imprecise concepts doesn't make them any more precise*.”

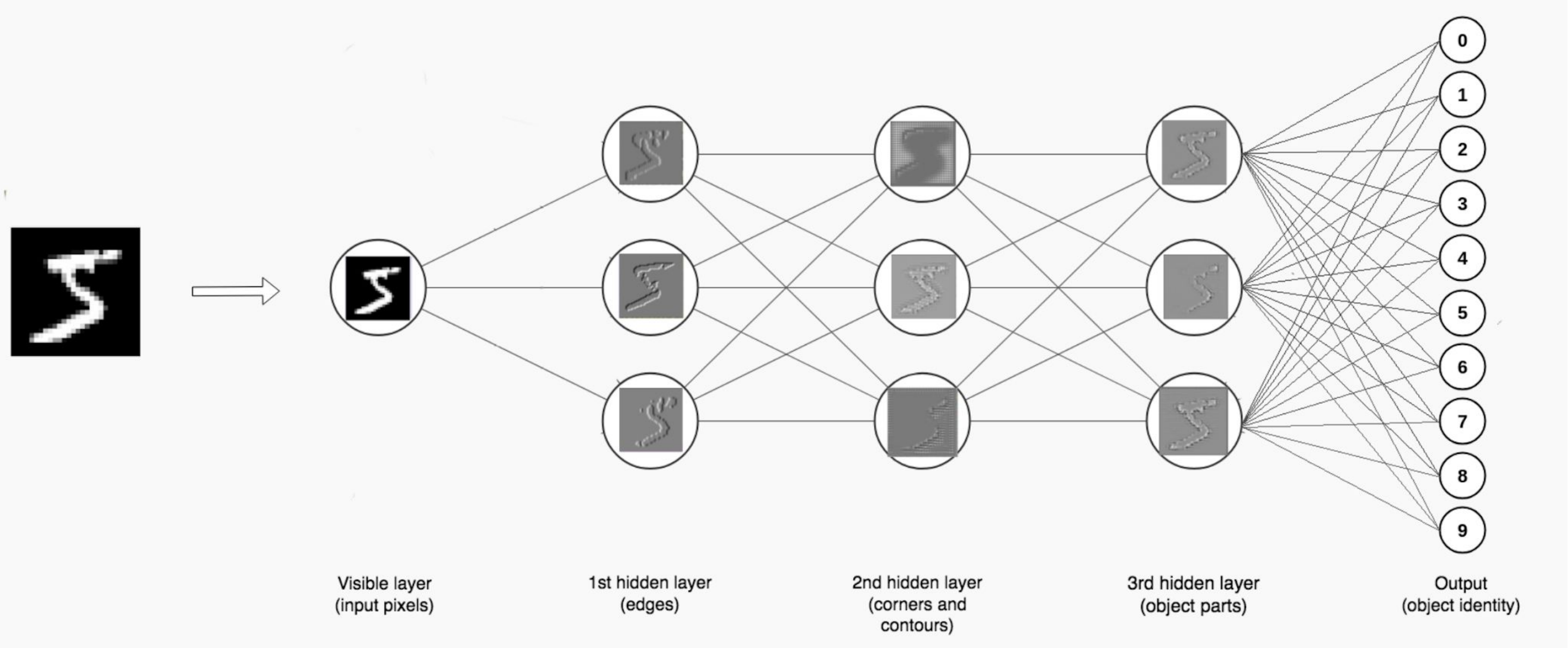
Logic rules need to represent all possibilities to be useful

Yale shooting problem:

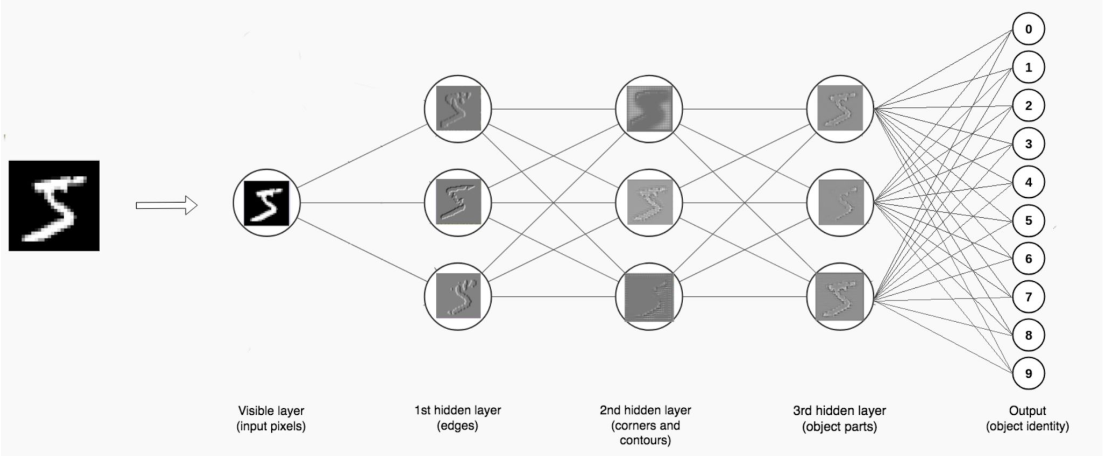
1. Fred is alive and Alice has a gun
2. Alice loads the gun
3. Alice shoots at Fred
   a. She missed
   b. She shot in different direction
   c. She hits in the arm
   d. They're not at the same location
   e. She thinks she shoots at Fred but it's a different Fred
   f. It's a toy gun
   g. The gun broke
   h. Fred was revived, life saved
   i. A dual
4. Fred is dead

If the first 3 events, is the 4th event true?

A modern neural net vs. the 1970s Perceptron



A modern neural net (2005+) vs. the 1970s Perceptron



The big differences:

- Non-linear functions + Multiple layers
- Lot more data!

Summary of history so far

ML has seen successive eras dominated by:

- Rule-based algorithms

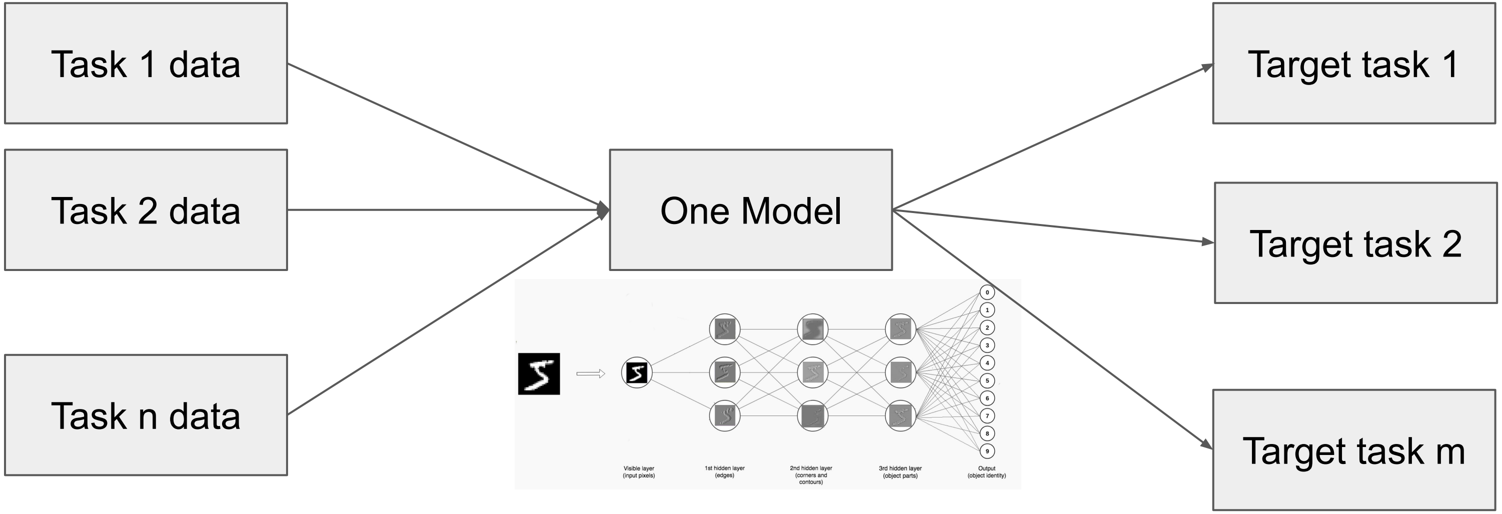
- Hand-crafted features

- Limited generalization

Enter Foundation models

- Large data-driven models
  - More, better data is the main reason models work better; not cleverer rules or algorithms
- Transfer learning
  - If done right, you can teach a model one task, and ask it to complete a different task
- Broad generalization
  - If you do enough transfer learning, you can generalize broadly to many tasks

Two approaches to building a generalized model



Task 1 data

Task 2 data

Task n data

One Model

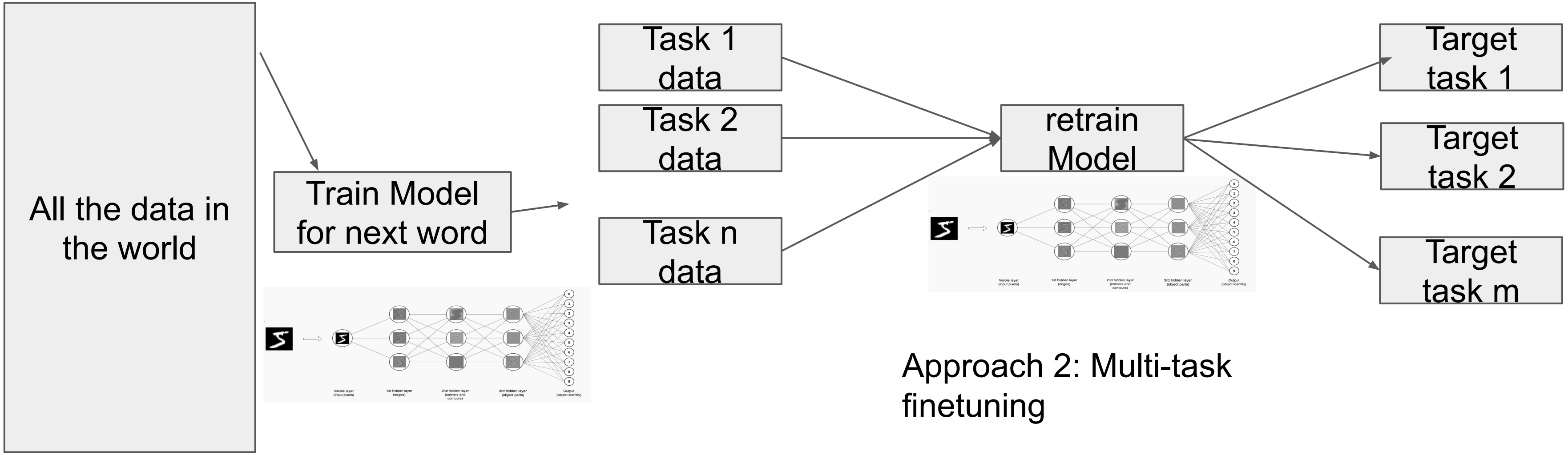
Target task 1

Target task 2

Target task m

Approach 1: Multi-task training

Two approaches to building a generalized model



All the data in the world

Train Model for next word

Task 1 data

Task 2 data

Task n data

retrain Model

Target task 1

Target task 2

Target task m

Visible layer (input pixels)
1st hidden layer (edges)
2nd hidden layer (corners and contours)
3rd hidden layer (object parts)
Output (object identity)
0
1
2
3
4
5
6
7
8
9

Approach 2: Multi-task finetuning

Which approach should work better? Why?

PLMs -> ILMs

PLMs are “pretrained language models”, which are usually trained to predict the next word or “token”.

Instruction tuned LMs (ILMs) go a step further. The take PLMs, and train them further on instructions.

E.g. “Write an essay on the civil war: ____”

Most LLMs you will use in this course are ILMs

ILMs are a better interface

- Easier to use: just say what you want
- Easier to “steer”: Many ILMs now understand “Do not do X”
- Many are safety-tuned: they are less likely to say offensive things (but this tuning is not perfect. Be prepared for surprises!)

- Chat ILMs were originally meant for conversations. But they are surprisingly versatile. I suggest you use them as “default”

LLM terminology

LLM instructions = “prompts”

LLM outputs = “predictions” (but outputs is also in common lingo)

LLM temperature = “how consistent do I want my outputs to be?”

0 = deterministic

Anatomy of a prompt

translate english to french:

Instruction

English: "I like apples"

French: « J'aime les pommes »

English: "I like the sun"

French: « J'aime le soleil »

Examples

English: " I like the rain"

French: « J'aime la pluie »

English: "Dinosaurs are cool"

French:

Types of prompts

Zero-shot: No examples, only instruction

Few-shot: a “few” examples. May not have an instruction.

How many examples?

- Rule of thumb: no more than 8.
- If you can’t do it in 6-8, the task is too complex / model is too simple, and you should try and change the task instead

Do you need examples?

- Often, instructions alone work better than examples + instruction.
    - “Bad” examples are worse than no examples

Bad examples

- Bad examples muddy the task

```
Classify the following reviews as helpful or not helpful:

"This was a life-saver! I don't know what I would have
done without it"-> Helpful
"I received the wrong item, and getting a refund was a
nightmare." -> Not helpful
"This is awful" -> Helpful
"I hate wasting money" -> Not helpful
```

Bad examples

- Bad examples shift the boundaries of what is acceptable

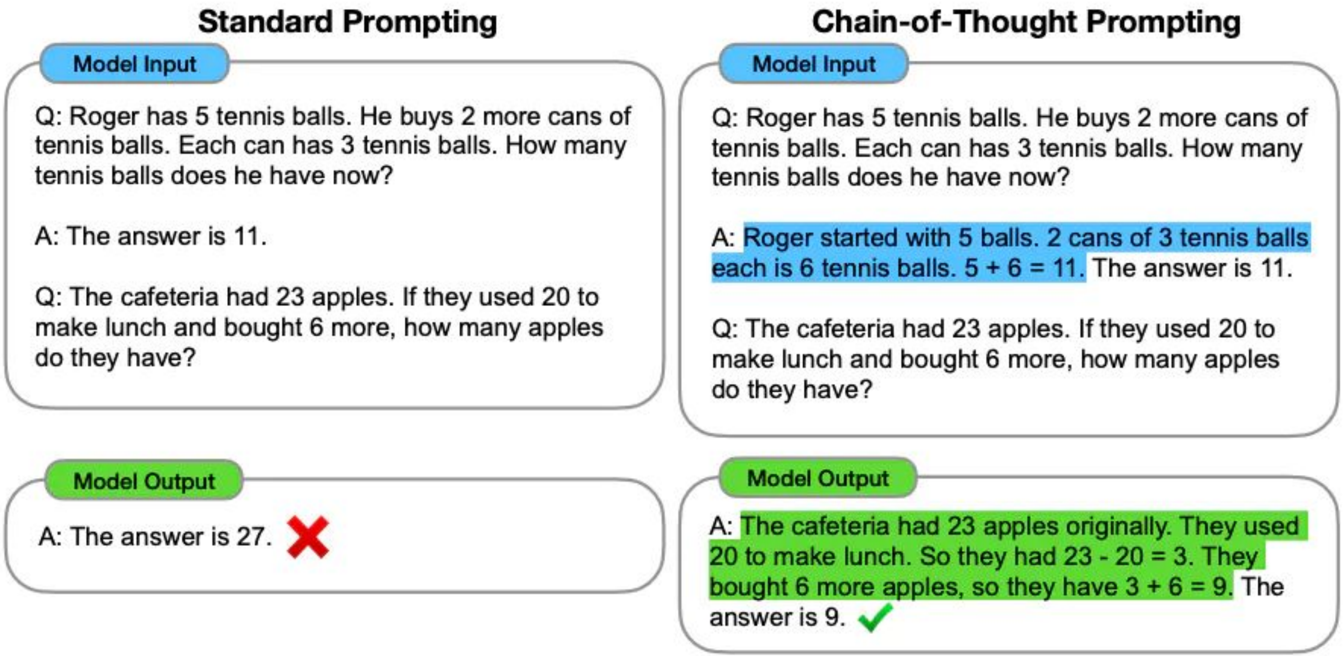
```
Classify the following reviews as helpful or not helpful:

"This was a life-saver! I don't know what I would have
done without it"-> Helpful
"I received the wrong item, and getting a refund was a
nightmare." -> Not helpful
"This is awful" -> Helpful
"I hate wasting money" -> Not helpful
"मुझे नहीं पता कि मैं इसके बिना क्या करता।" -> Helpful
```

Good examples

- Good examples help clarify what is *typical,* not what is the edge-case
- Good examples are precise in following instructions (if any)

Bottomline: if you use vague examples, you teach the model to be “lazy”.

Chain of thought prompting



Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Multimodal Models: Beyond Text

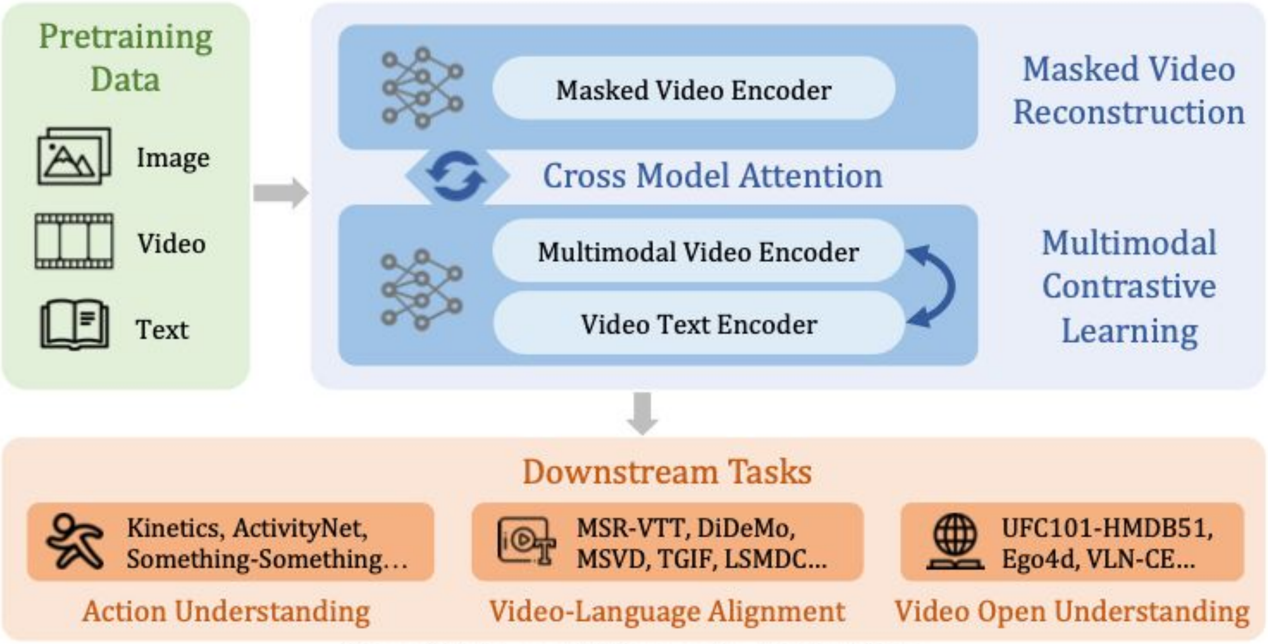


Figure 2: The overall framework of InternVideo.

https://arxiv.org/abs/2212.03191

What’s next for models? One guess: Continuous Learning, Adaptation, Grounding

ChatGPT training data collected before Sept 2021

- The world has changed since then
- How do we use new data usefully?

Adaptation:

- “How do you bake a cake?” <- What happens when user tells you it’s too sweet?

Grounding:

- Models so far don’t really know what’s happening outside their data.