Introduction

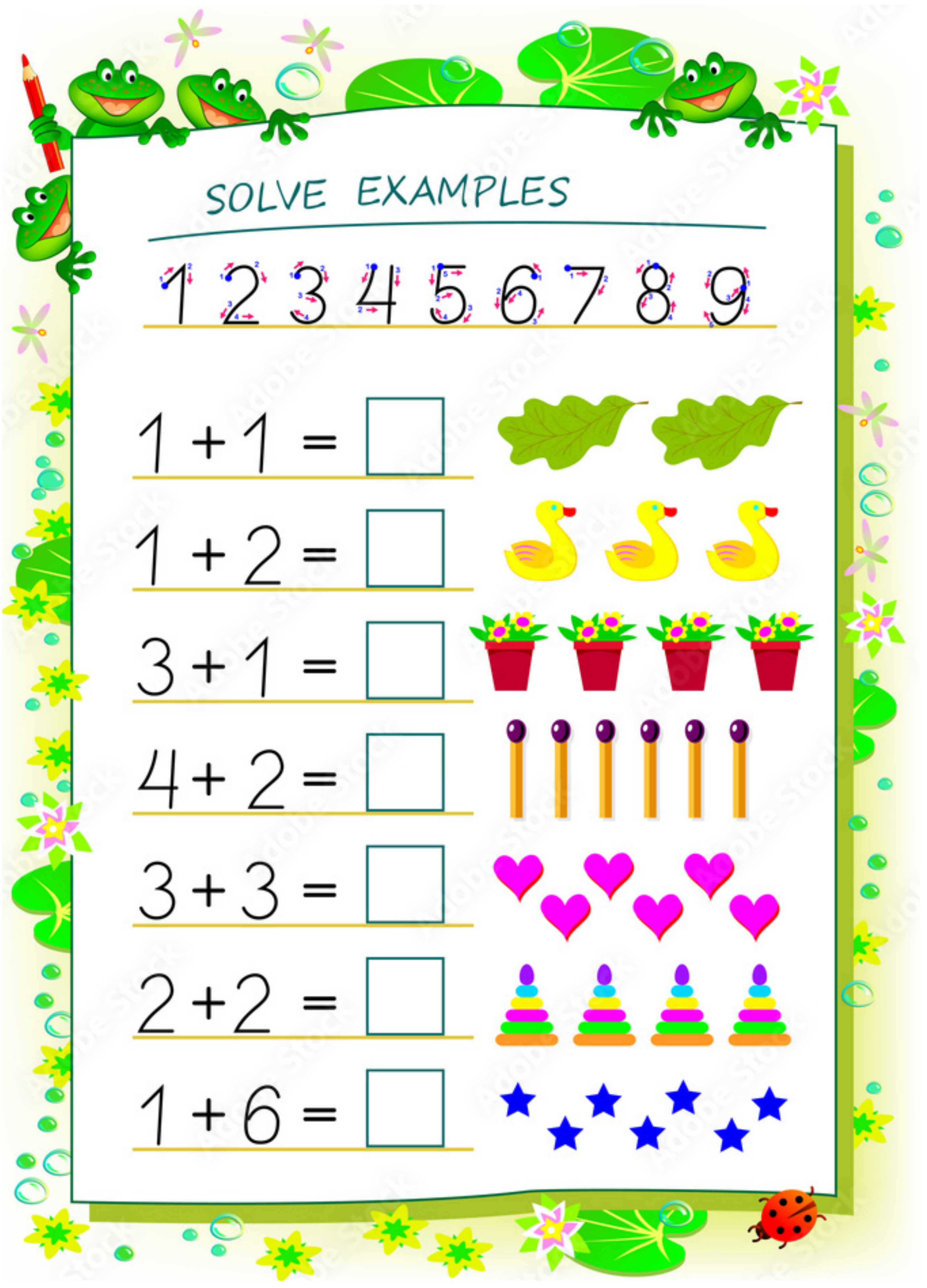
EECE454 Intro. to Machine Learning Systems

Fall 2024

What is machine learning?

Human Learning

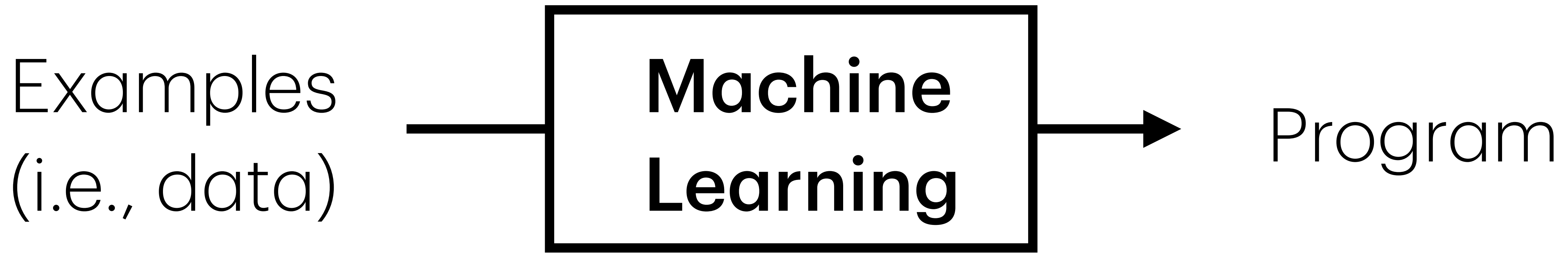
- How do human learn?
  - Given some **examples**, human can **find a pattern**.



Human Learning

- How do human learn?
    - Given some **examples**, human can **find a pattern**.

- **Machine Learning.** Given some examples, a machine automatically:
    - discovers some pattern from the examples
    - builds some program that utilizes such discovered pattern.



Human Learning

- How do human learn?
    - Given some **examples**, human can **find a pattern**.

- **Machine Learning.** Given some examples, a machine automatically:
    - discovers some pattern from the examples
    - builds some program that utilizes such discovered pattern. ← **What do we mean?**

Examples (i.e., data) → **Machine Learning** → Program

Example tasks

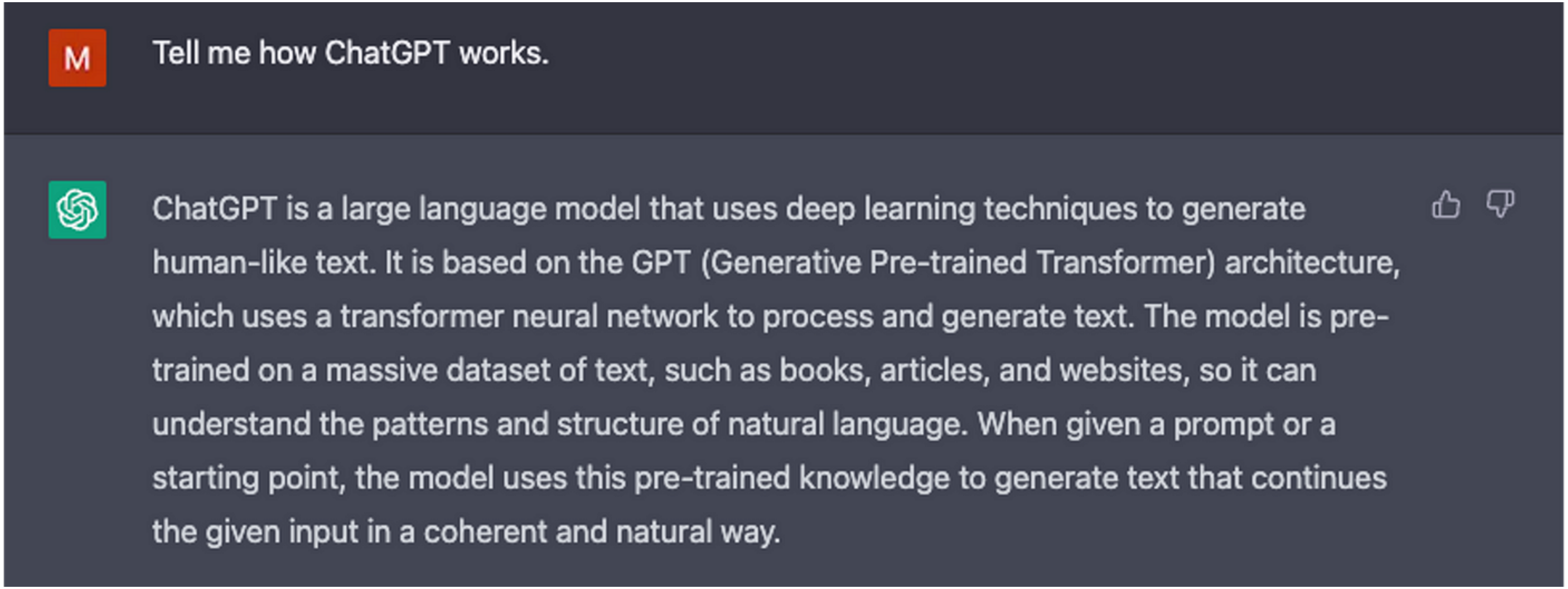
- Create a program that, given **an image of a dog**, returns the **name of the dog specie**
    - Human will need a lot of (image, species) pairs

Example tasks

- Create a program that, given **a Netflix user** and **a movie**, returns the **expected user rating**
    - Human will need a lot of (user, movie, rating) triplets

Example tasks

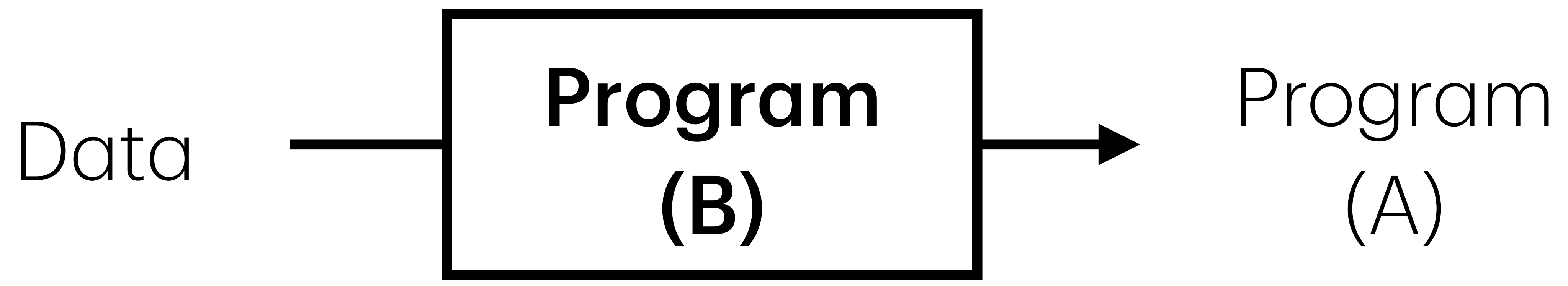
- Create a program that, given **a text input**, returns a **human-like response** (or better)
  - What data do we need?



Terminologies

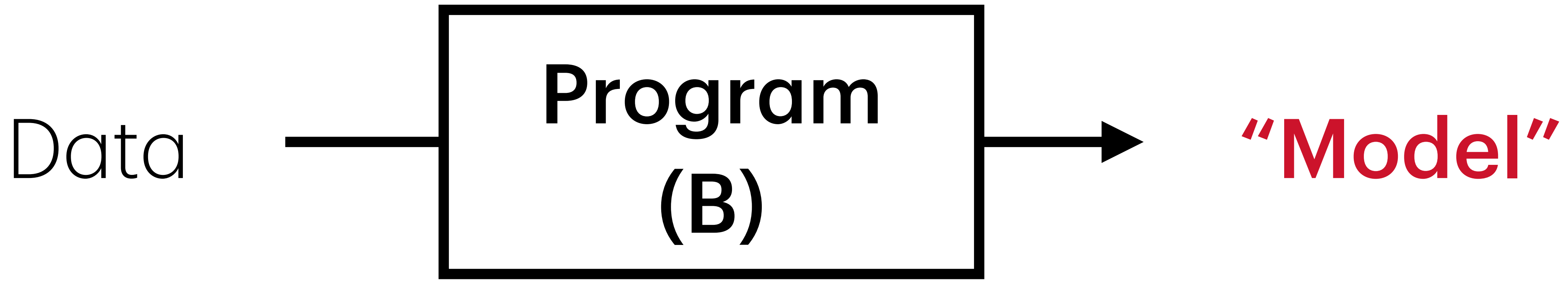
Terminologies

- Notice that there are **two programs in action**
  - (A) The program that utilizes the pattern
  - (B) The program that discovers patterns from data to build (A)



Terminologies

- Notice that there are **two programs in action**
  - (A) The program that utilizes the pattern
  - (B) The program that discovers patterns from data to build (A)



- The program (A) is called **“model”** (or “predictor” or “hypothesis”) and what (A) does is called **“prediction”** (or “inference”)

Terminologies

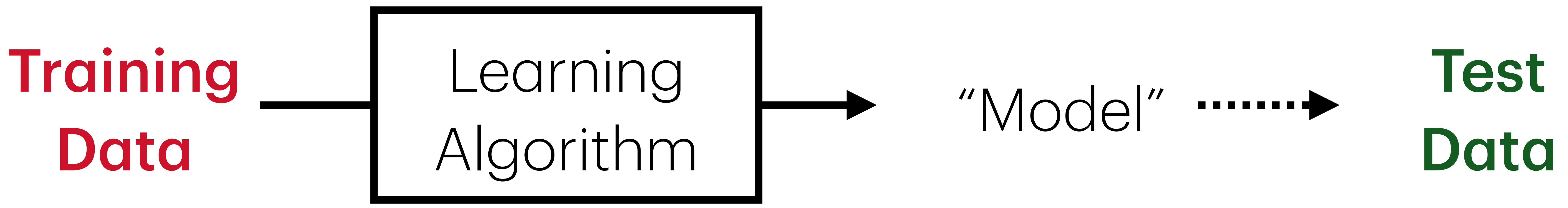
- Notice that there are **two programs in action**
  - (A) The program that utilizes the pattern
  - (B) The program that discovers patterns from data to build (A)

- The program (A) is called **"model"** (or "predictor" or "hypothesis") and what (A) does is called **"prediction"** (or "inference")

- The program (B) is called **"learning algorithm"** and what (B) does is called **"training"**

Terminologies

- Notice that there are **two programs in action**
    - (A) The program that utilizes the pattern
    - (B) The program that discovers patterns from data to build (A)



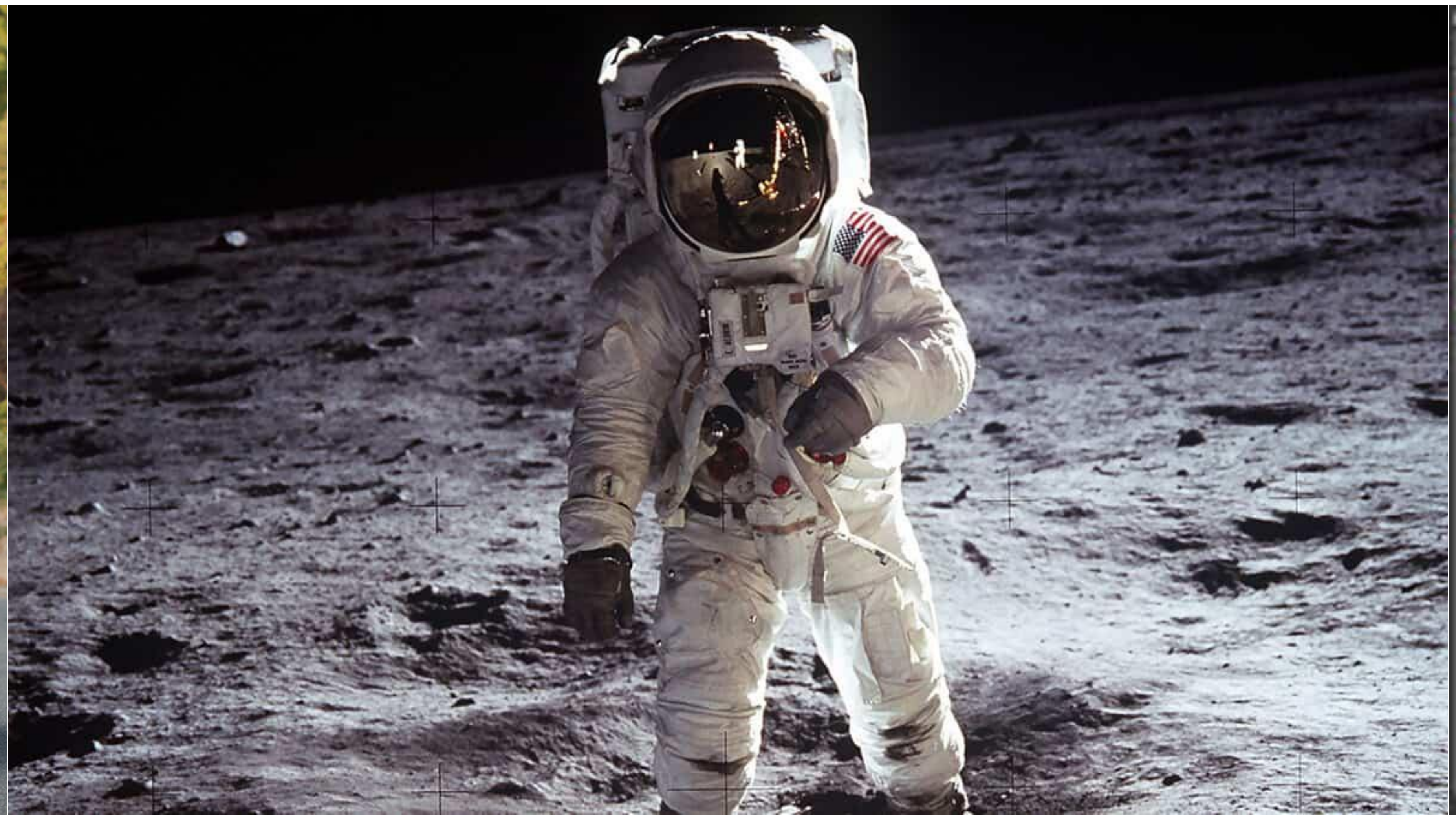
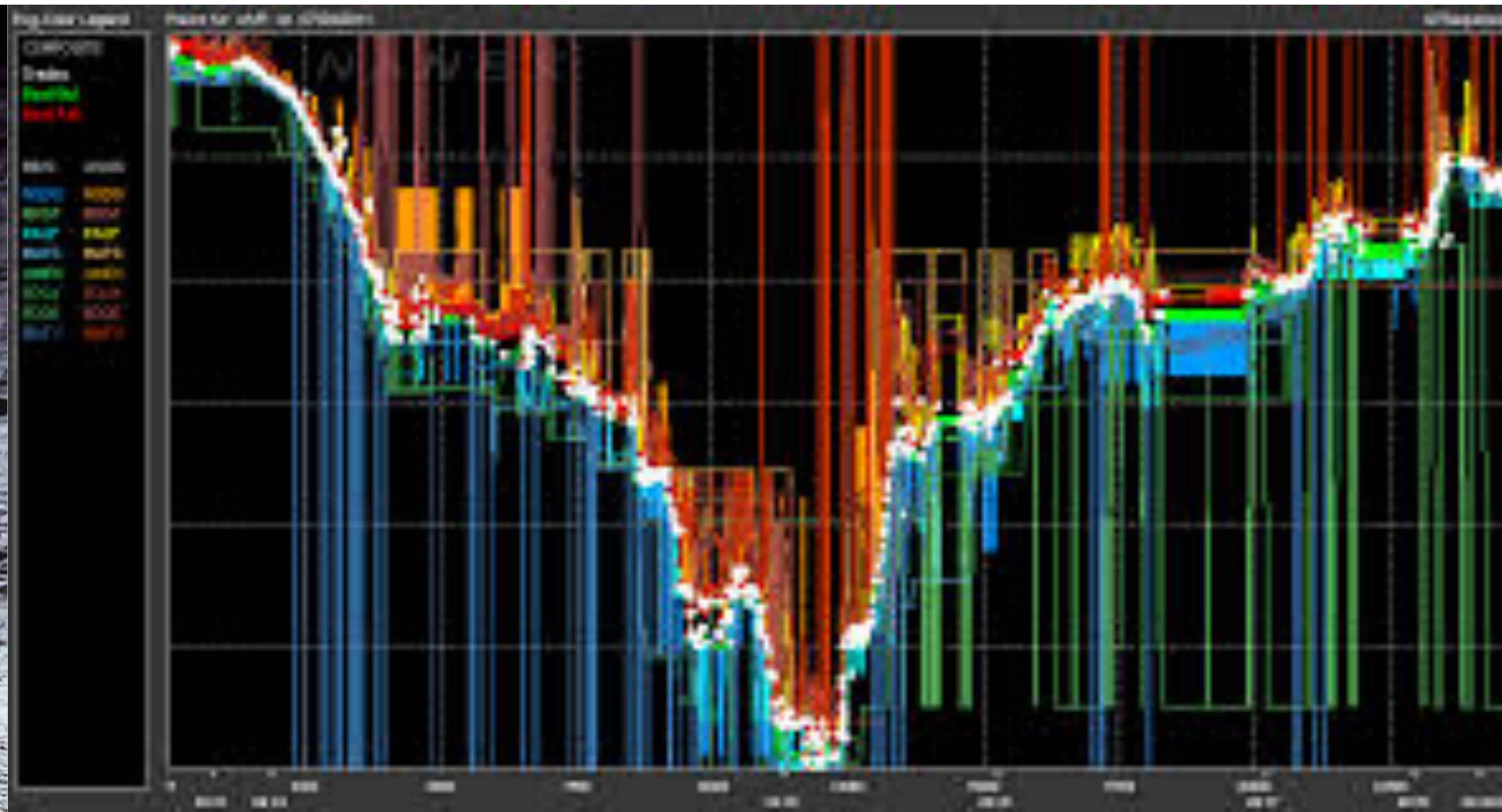
- The learning algorithm sees the **training data**,
  whereas the model will be used on a new, incoming data, called the **test data**.

(otherwise, we call it "data mining")

Why “machine” learning?

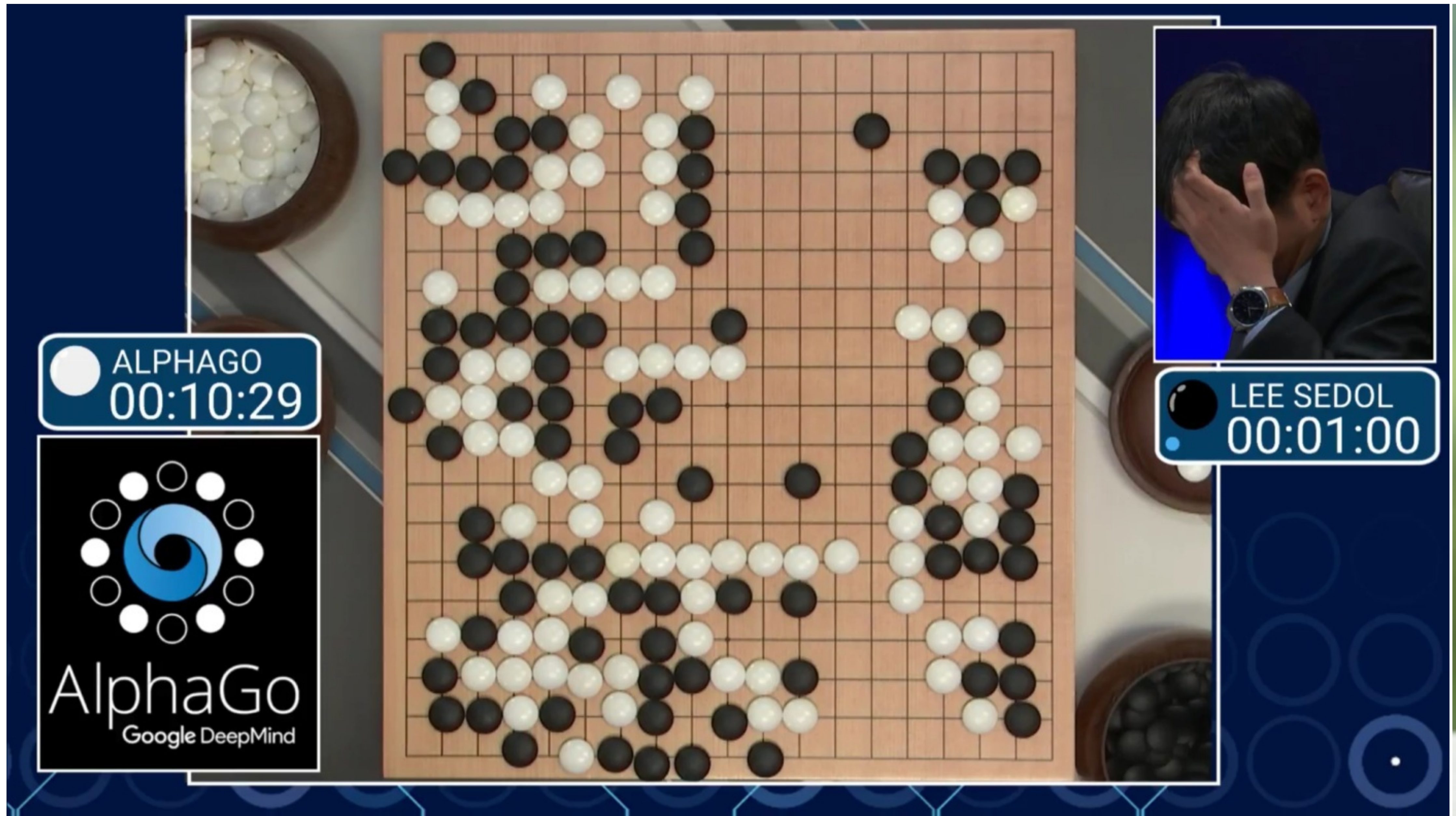
Why machines?

- We want machines to **use the patterns (prediction)**, because…
  - Human attention is limited (e.g., self-driving cars)
  - Human are vulnerable (e.g., space mission)
  - Human are slow (e.g., high-frequency trading)
  - Human are expensive (e.g., chatbots)

Why machines?

- We want machines to **find the patterns (training)**, because…
  - Human are dumb (e.g., AlphaGo)
  - Dataset is too big to handle (e.g., machine translation)
  - Difficult to write a code that uses human knowledge (e.g., dog classification)



What do “we” do for ML?

So what does human do?

- **ML algorithm researchers** develops the learning algorithm that
  - can train a model that solves a new task
  - requires very small computational cost
  - needs very small data to learn good models

So what does human do?

- **ML algorithm researchers** develops the learning algorithm that
  - can train a model that solves a new task
  - requires very small computational cost
  - needs very small data to learn good models
- **ML theory researchers** develop mathematical frameworks that can
  - give performance guarantees on ML algorithms and models
  - inspire new algorithms

So what does human do?

- **ML algorithm researchers** develops the learning algorithm that
  - can train a model that solves a new task
  - requires very small computational cost
  - needs very small data to learn good models
- **ML theory researchers** develop mathematical frameworks that can
  - give performance guarantees on ML algorithms and models
  - inspire new algorithms
- **ML system researchers** develop efficient systems for running ML algorithms and models

How do ML algorithms work?

How do ML algorithms work?

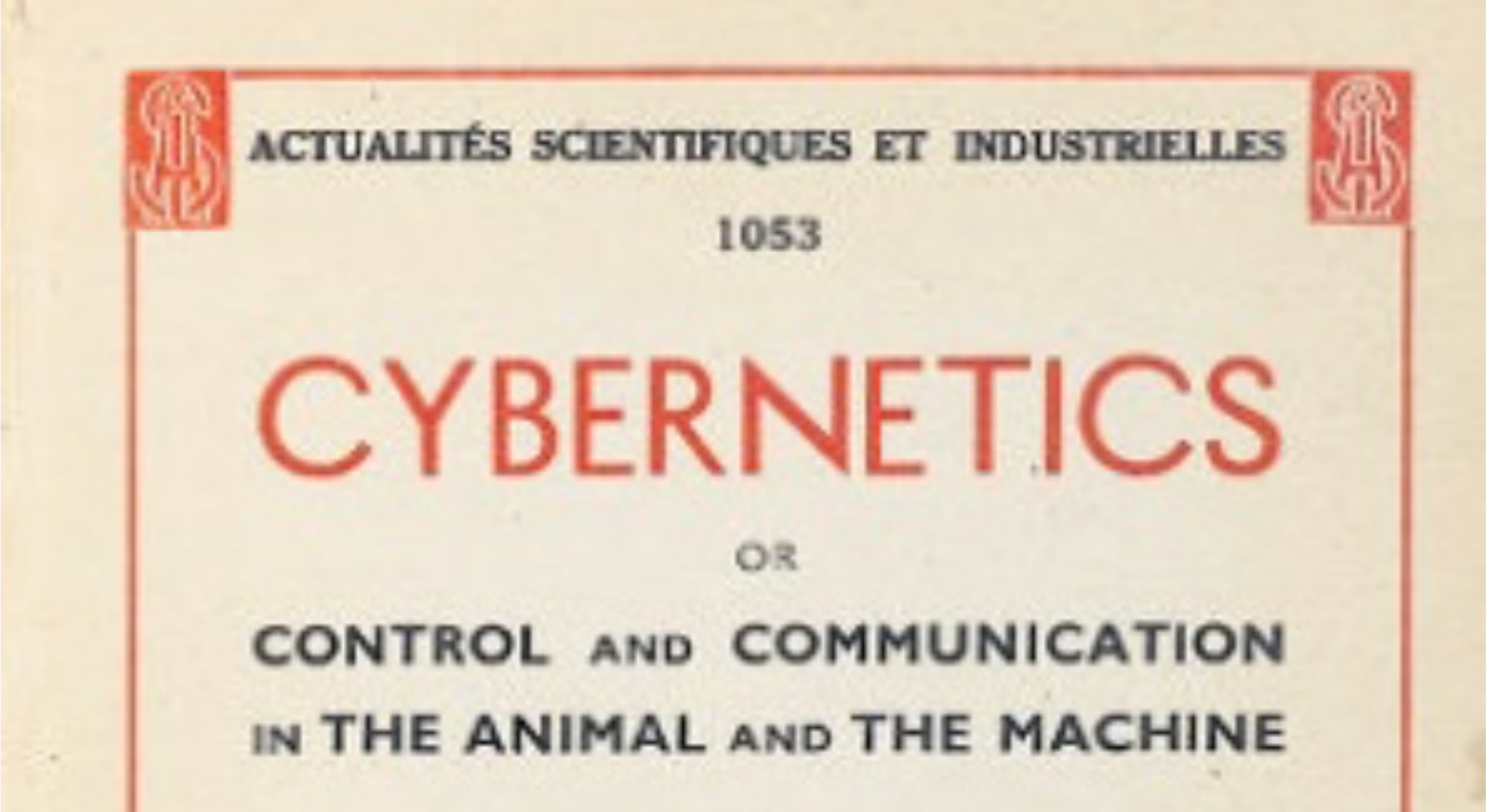
- Very difficult to give a simple answer
    - There are so many ML algorithms
    - They all work quite differently from each other
    - Some work well for this, some work well for that

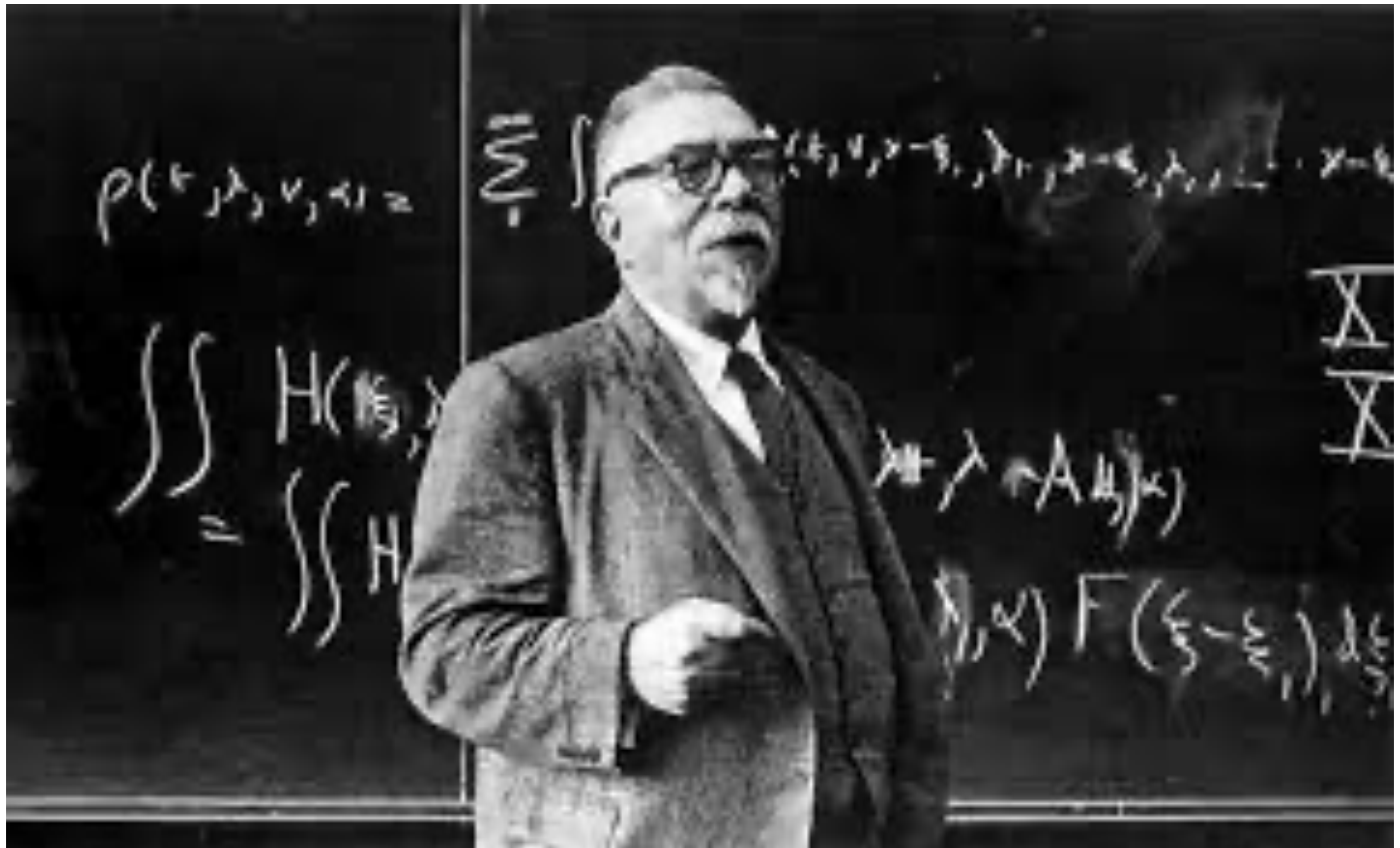
How do ML algorithms work?

- Very difficult to give a simple answer
    - There are so many ML algorithms
    - They all work quite differently from each other
    - Some work well for this, some work well for that
- **Today.** Very briefly discuss two unifying perspectives.
    - "Cybernetics" paradigm
    - "Statistical Learning" paradigm
- There are competing paradigms, of course, e.g., Bayesian ML.

Cybernetics (1947)

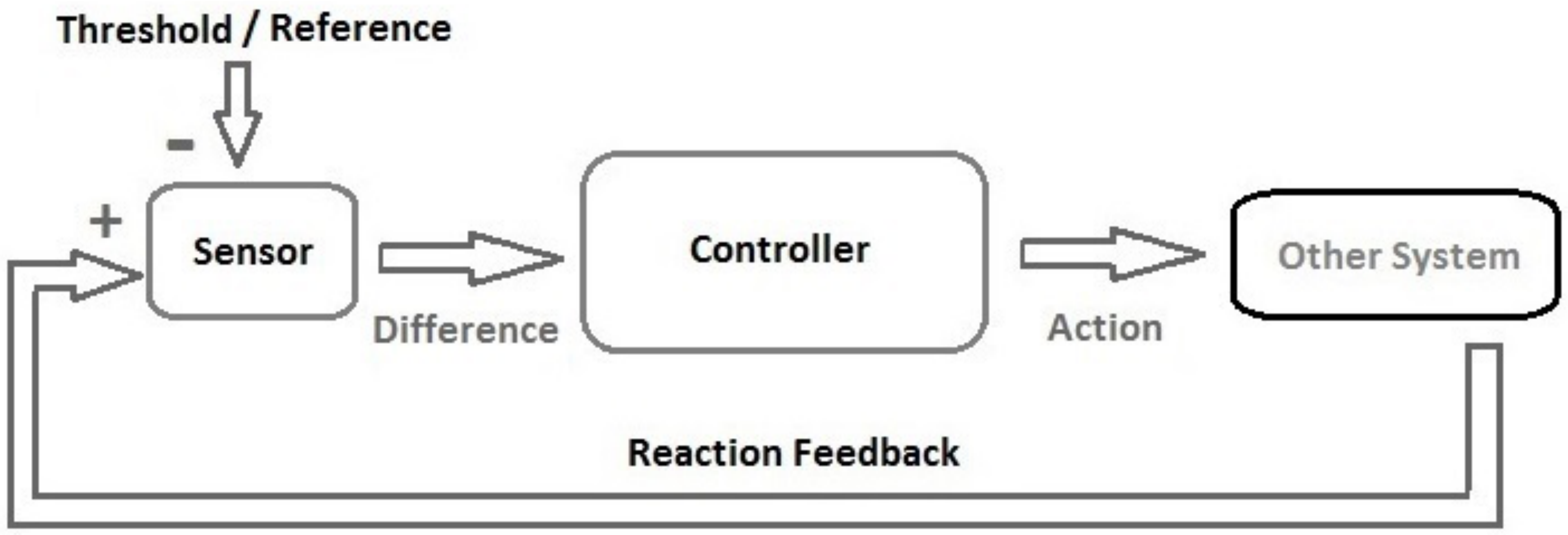
- **Cybernetics.** The origin of "artificial intelligence"
  - Coined by a control theorist Norbert Wiener





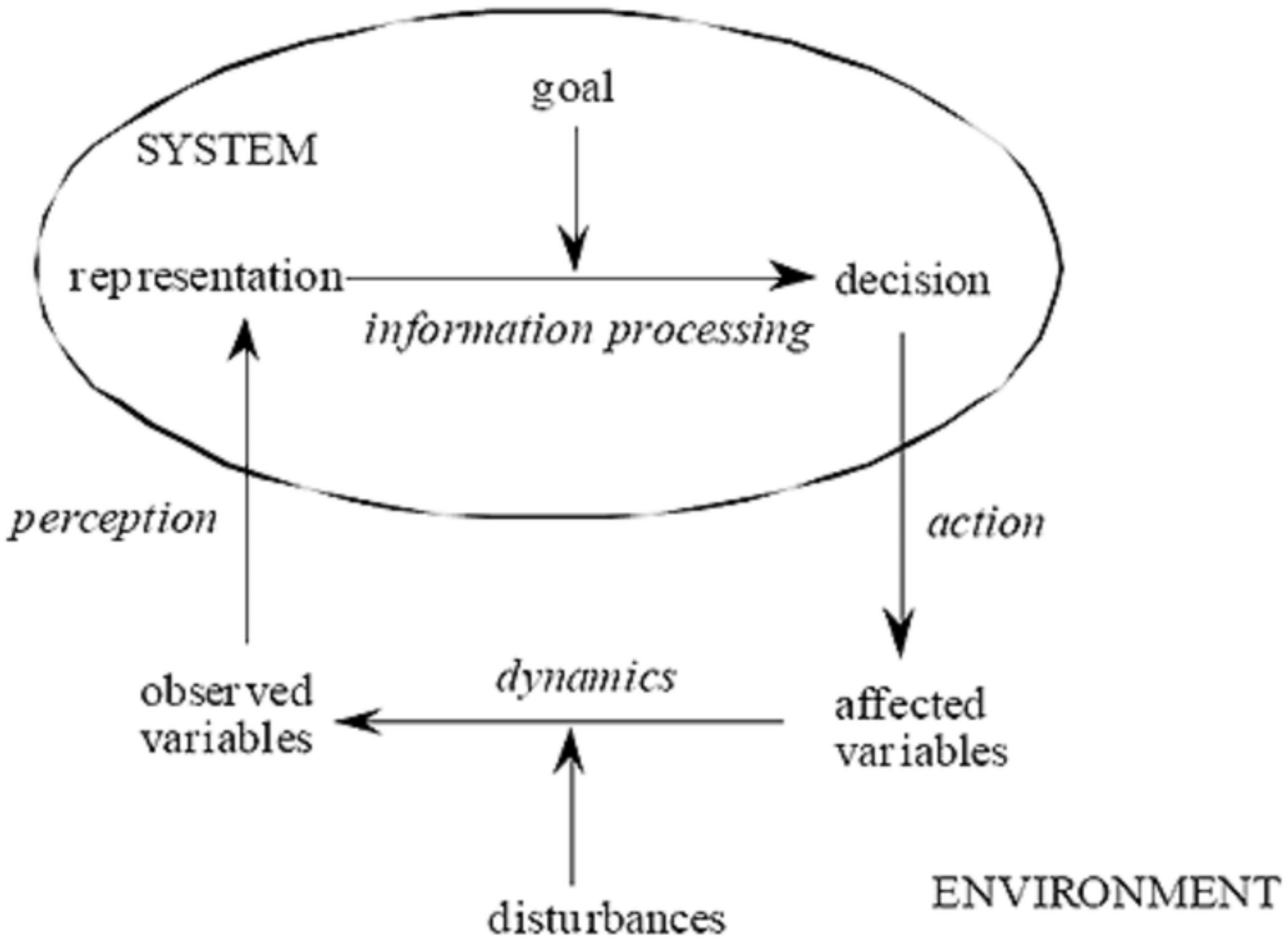
Cybernetics (1947)

- **Cybernetics.** The origin of "artificial intelligence"
  - Coined by a control theorist Norbert Wiener
- Wiener viewed intelligence as a **"circular causal process, via feedback loop"**
  - thus called "κυβερνήτης" (steering)
  - proposed a holistic study of communication, control, and feedback mechanisms.



Cybernetics (1947)

- Cybernetics provided all core concepts.
  - We have some **model** with **changeable internal states** (i.e., model parameters)
  - We find the **right internal state** (i.e., optimize) by repeating
    - Test the current program on training data
    - Get the feedback
    - Modify the state accordingly
- Exactly what modern ML or RL does!



Statistical Learning (1968)

- **Statistical Learning.** Followed the cybernetics rush in Soviet union (Lyapunov, Kolmogorov, ...)
  - Core ideas developed by Vladimir Vapnik and Alexey Chervonenkis

NEWS

Prestigious AI Series Wraps up with Lecture by the Father of Machine Learning

POSTED:
MAY 14, 2018

Vladimir Vapnik, a professor at Columbia University's Center for Computational Learning System and Professor Anna Choromanska

Statistical Learning (1968)

- **Statistical Learning.** Followed the cybernetics rush in Soviet union (Lyapunov, Kolmogorov, ...)
  - Core ideas developed by Vladimir Vapnik and Alexey Chervonenkis

- An ML algorithm is defined by:
  - a **hypothesis space** (i.e., a bag of models)
  - a **loss function**
    - measures how bad a model is,
      when evaluated on a single example
  - a **search algorithm** to find the minimum loss model in this space
    - can be done by optimizing the internal parameters (can be NP-hard!)

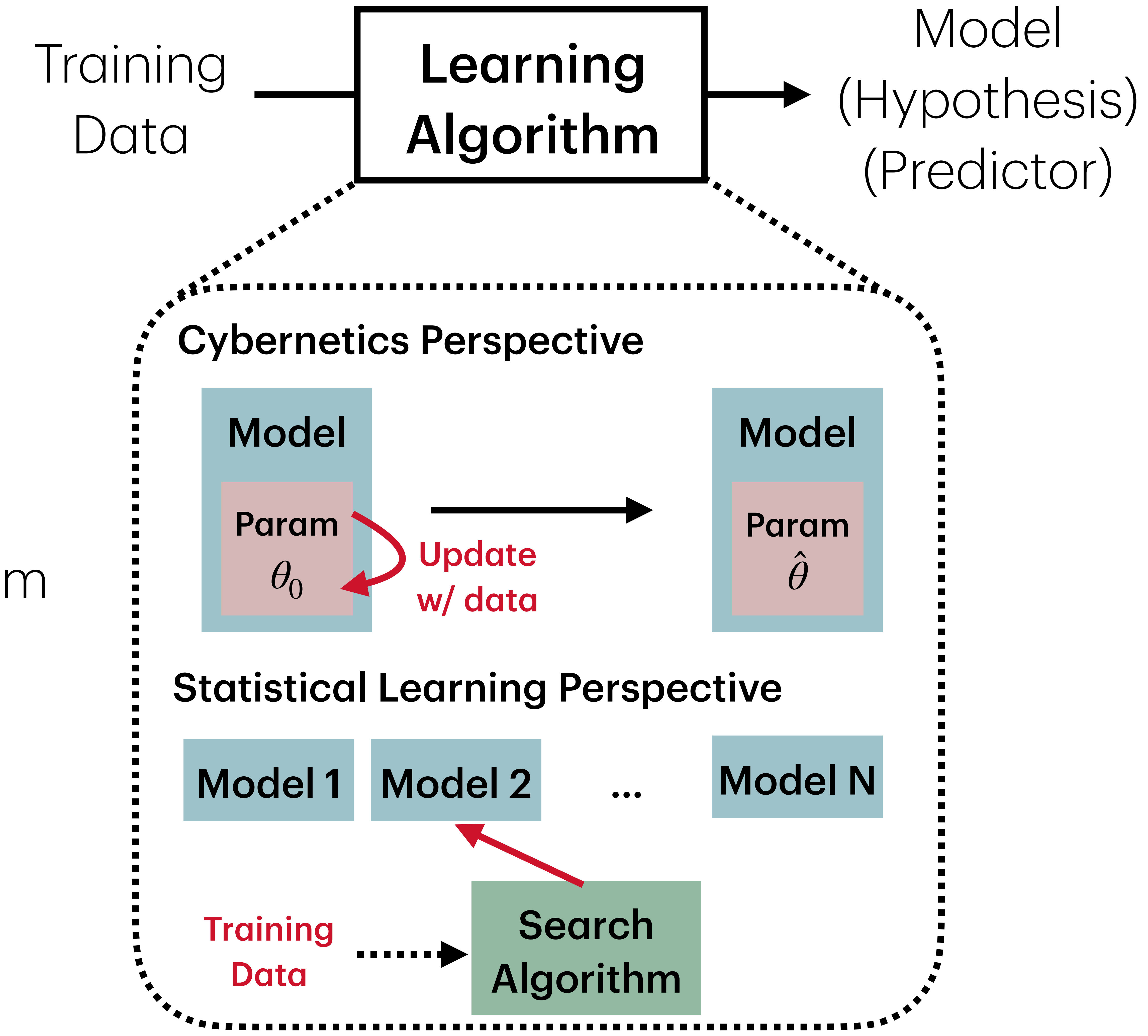
Two perspectives

- Common to both paradigms, we assume
  - Access to some data $Z_1, \ldots, Z_n$ (either sequentially or available as a batch)
  - Access to some hypothesis space

$$\mathcal{F} = \{f_1, f_2, \ldots\}$$

  - The ML algorithm solves an optimization problem to minimize the loss on the data

$$\min_{f \in \mathcal{F}} \ell(f, (Z_1, \ldots, Z_n))$$



Two perspectives

- Different from usual optimization literature, ML is about **generalization to new data**
  - That is, we reduce the empirical error

$$\min_{f \in \mathcal{F}} \ell(f, (Z_1, \ldots, Z_n))$$

  and yet, we want the solution $\hat{f}$ to work well on new data, i.e., have a small

$$\mathbb{E}[\ell(\hat{f}, Z_{\text{new}})]$$

  - This is the defining characteristic of ML frameworks.
    - This also makes the field highly empirical; we know very little about the distribution of Z.

What does this course do?

This course teaches …

- This course consists of two parts:
  - Part 1. Introduce classic ML frameworks
  - Part 2. Familiarize you with basics of deep learning (+ hands-on experience)

This course teaches …

- This course consists of two parts:
  - Part 1. Introduce classic ML frameworks
  - Part 2. Familiarize you with basics of deep learning (+ hands-on experience)
- **Why learn classic frameworks?**
  - Outperforms DL in many tasks (e.g., tabular, time-series)
  - Have inspired or directly employed in many DL algorithms (e.g., VQ-VAE)
  - Neat to analyze; gives you strong understanding and intuition.

This course teaches …

- By the end of this course, I expect you to
  - be able to apply existing ML algorithms on real-world tasks
  - design your own ML frameworks and algorithms
  - conduct basic analysis on your algorithm and model

Administrivia

Team

- **Instructor.** Jaeho Lee 이재호
  - Assistant Professor @ POSTECH EE (2022.03 ~)
    Research Scientist @ Google (2023.09 ~)
  - jaeho.lee@postech.ac.kr
  - Responsible for: Coursework-related, Anything else.
- **TA.** Minjae Park 박민재
  - Ph.D. track @ POSTECH EE (2024.03 ~)
  - mjae.park@postech.ac.kr
  - Responsible for: Assignments, Grading, Attendance

Location & Hours

- **Class.** Engineering Building #3, Classroom 115
    - Mondays / Wednesdays 9:30AM — 11:00AM
- **Office hours.** Engineering Building #2, Office 323
    - Wednesdays 04:00PM — 05:00PM (+ by appointment)
- **Web.** https://jaeho-lee.github.io <— for lecture notes
  PLMS <— for assignment submissions

Grading

- Attendance: 10%
- Assignments: 30%
- Mid-Term: 30%
- Final Project: 30%

    - Graduate students will be graded separately
    - QE sit-ins will be judged based on how UGs do.

Prerequisites

- Not 100% required, but I assume you know:
  - Calculus
  - Basic linear algebra
  - Basic probability & statistics
  - Signals & Systems
  - Programming & Python

Textbook

- **Main**
    - “Mathematics for Machine Learning” by Deisenroth, Faisal, and Ong
        - https://mml-book.github.io
    - “Understanding Deep Learning” by Simon Prince
        - https://udlbook.githu.io/udlbook/

Textbook

- **Further Readings**
    - “Patterns, Predictions, and Actions” by Hardt and Recht
        - https://mlstory.org
    - “Dive into Deep Learning” by Zhang, Lipton, Li, and Smola
        - https://d2l.ai
        - Very recommended for programming exercises

Honor Codes

- **Simple principle: Cheating = F**
    - Sharing solutions —> not okay
    - Copying solutions —> not okay
    - Discussion —> do this with me or TA?
    - ChatGPT —> please don’t

Coming next

- We do some recap:
  - Linear Algebra
  - Optimization and Probability

Cheers