0. PROBLEMS
1. RULES
2. LEARNING

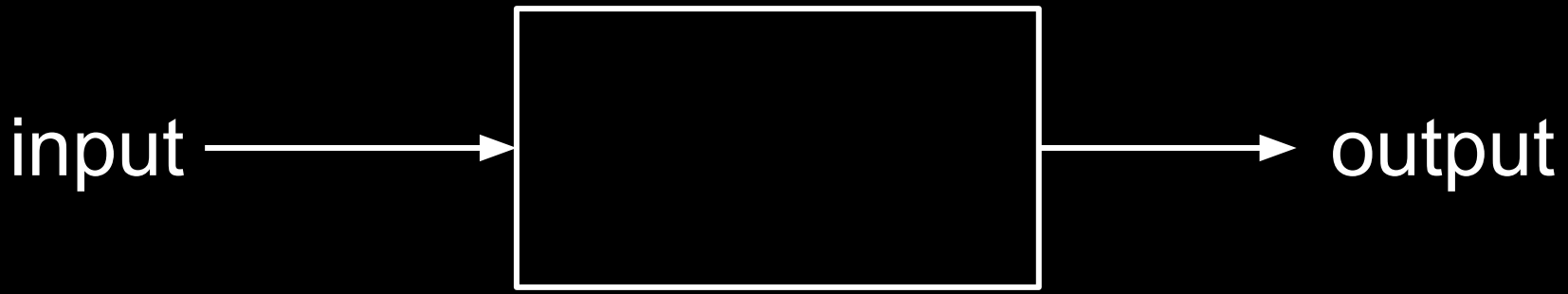
The slides are meant as visual support for the lecture.
They are neither a documentation nor a script.

Please consider the environment before printing the slides.

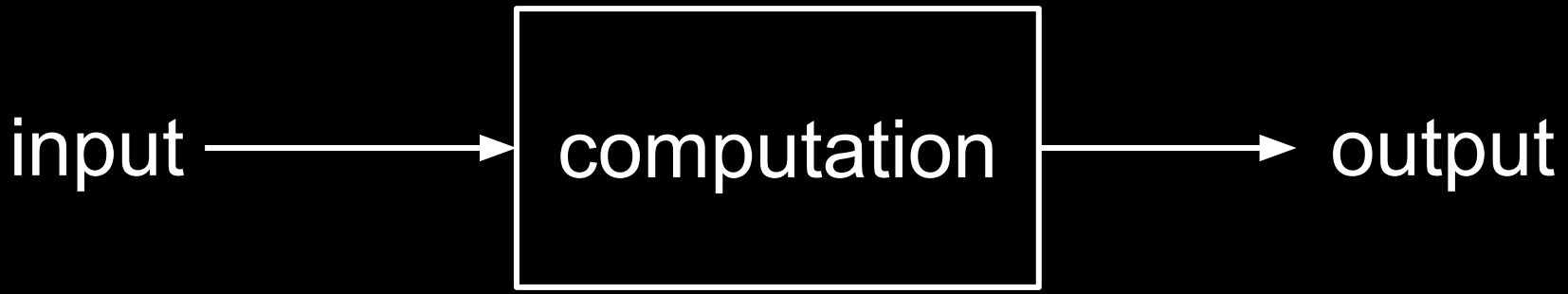
Comments and feedback at n.meseth@hs-osnabrueck.de

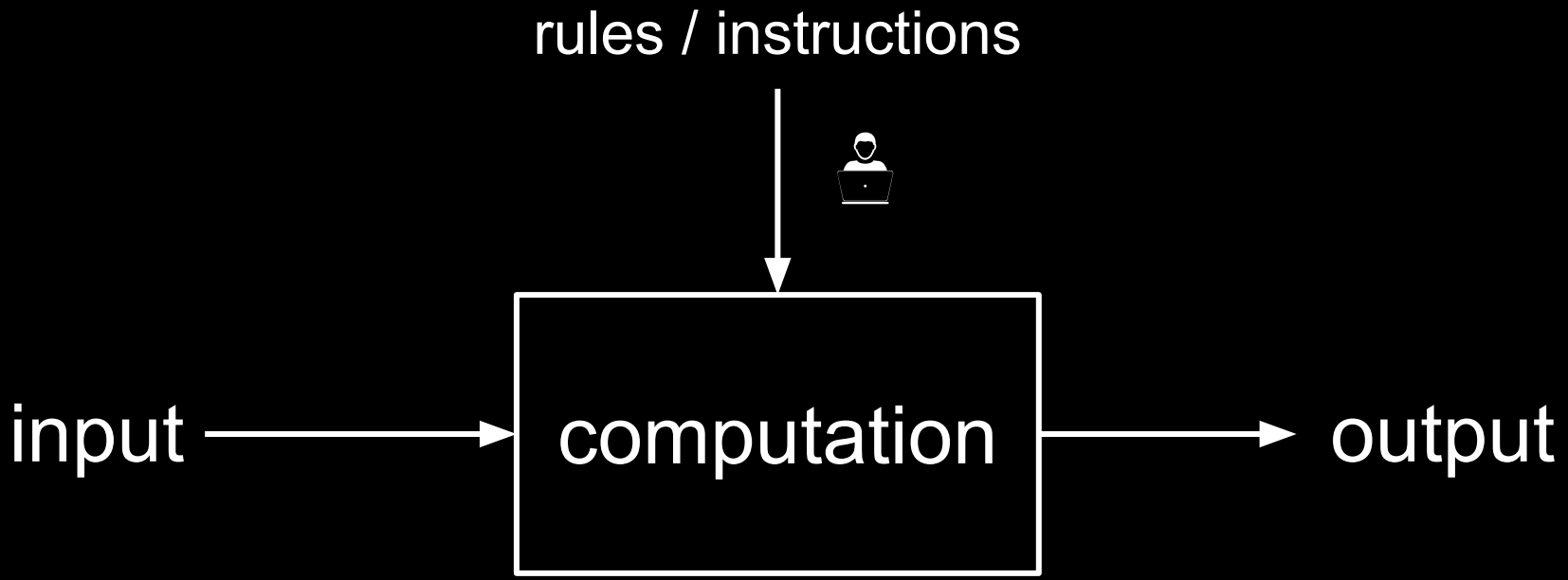
PROBLEMS

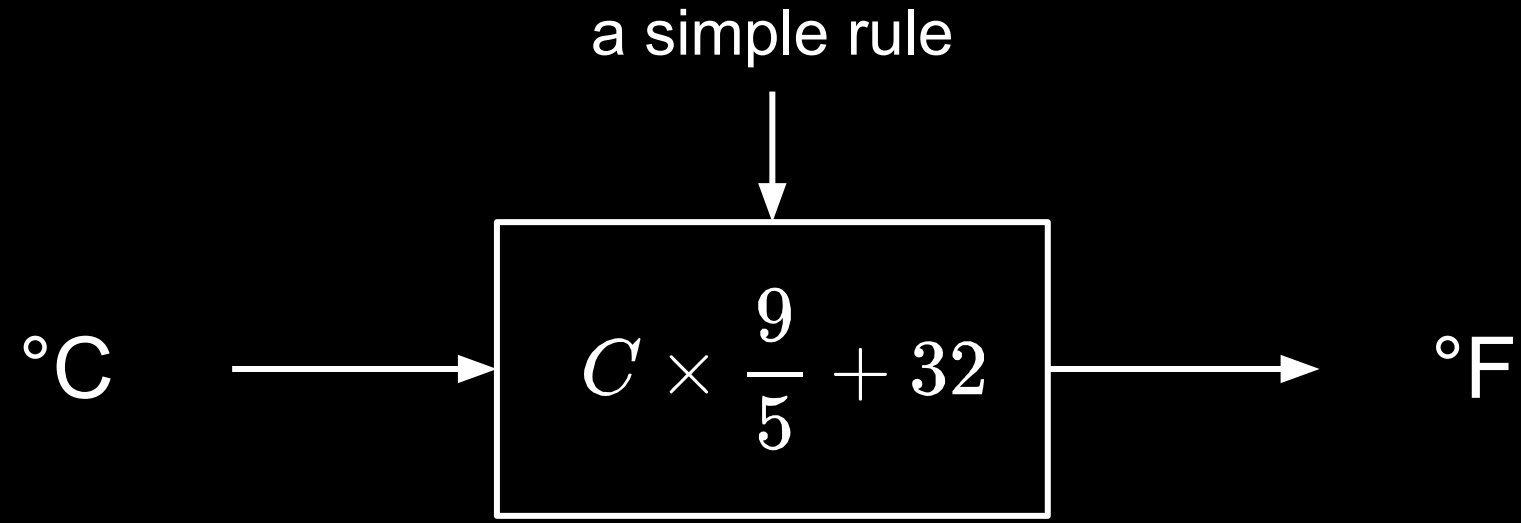
a model for solving problems

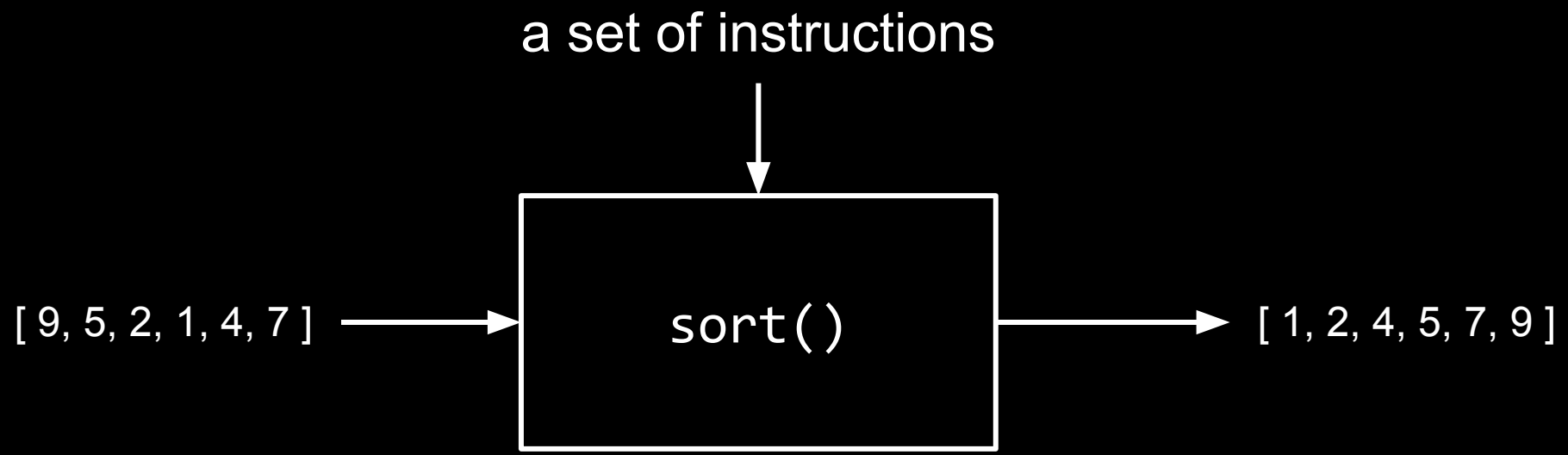


a model for solving problems




rules / instructions
input
computation
output

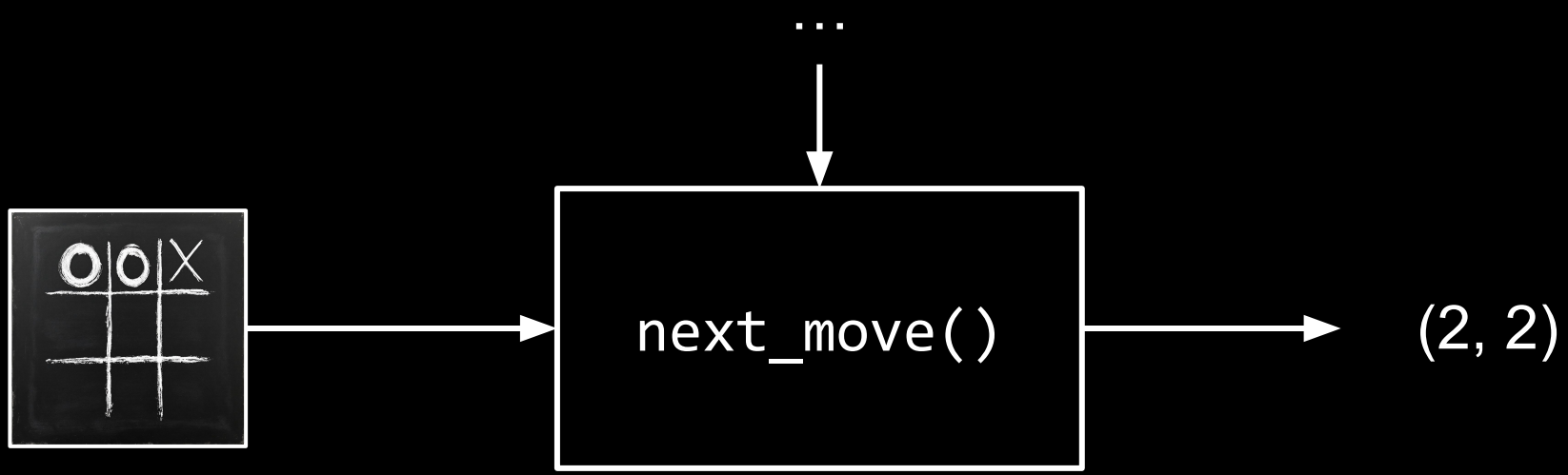

a simple rule
°C
$C \times \frac{9}{5} + 32$
°F


a set of instructions
[9, 5, 2, 1, 4, 7]
sort()
[1, 2, 4, 5, 7, 9]

`selection sort`:

find the smallest element and move it to front.

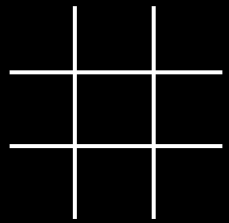
repeat for the rest of the elements.

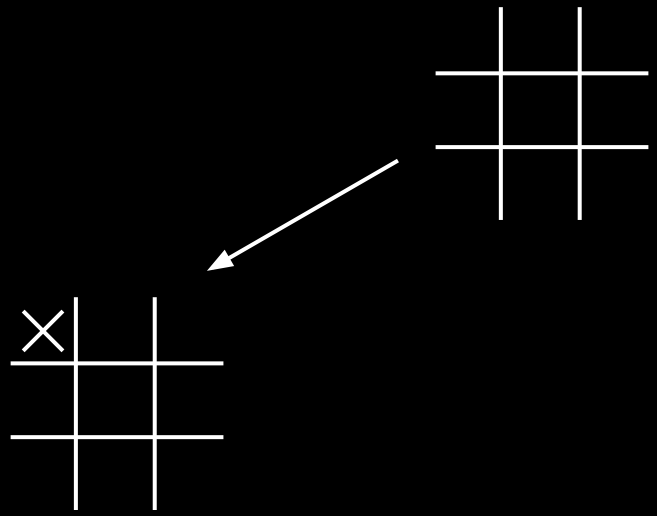

...
next_move()
(2, 2)

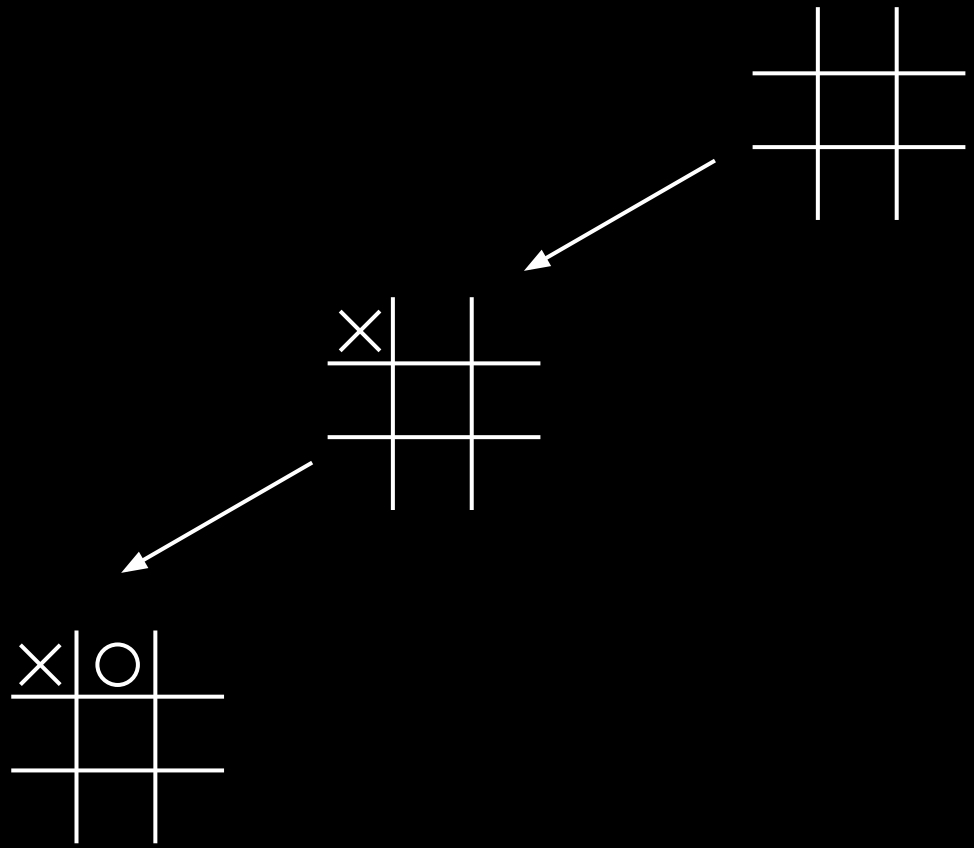
`brute-force search`:

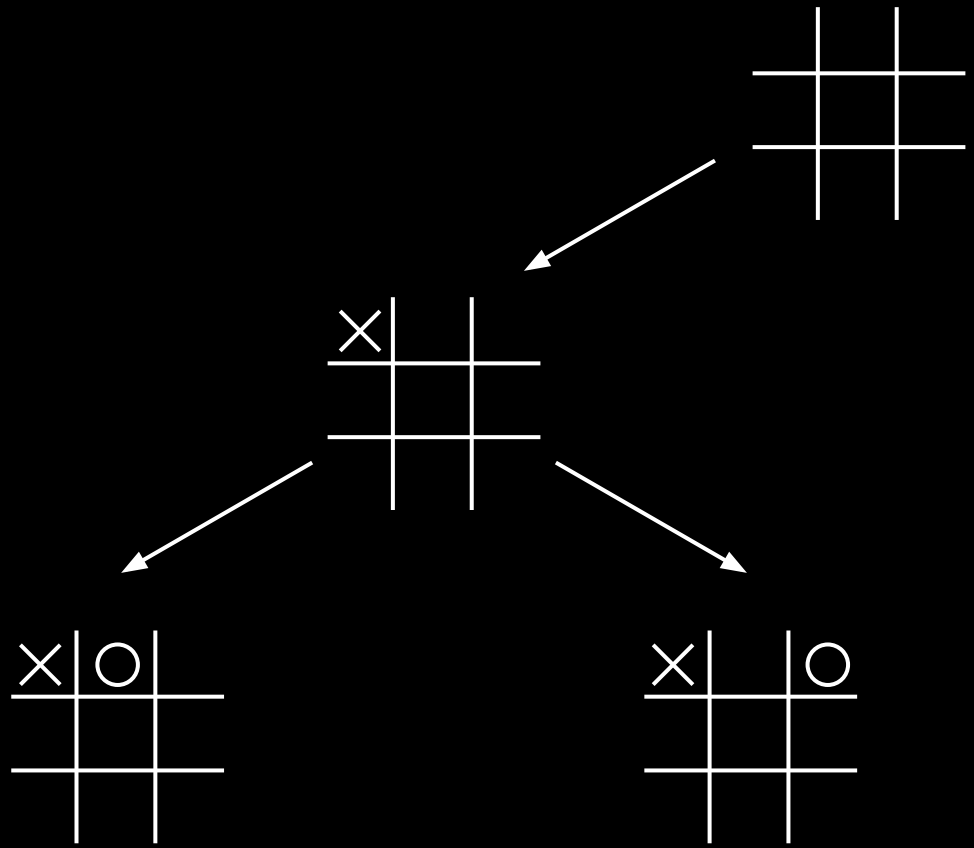
evaluate all possible move sequences.

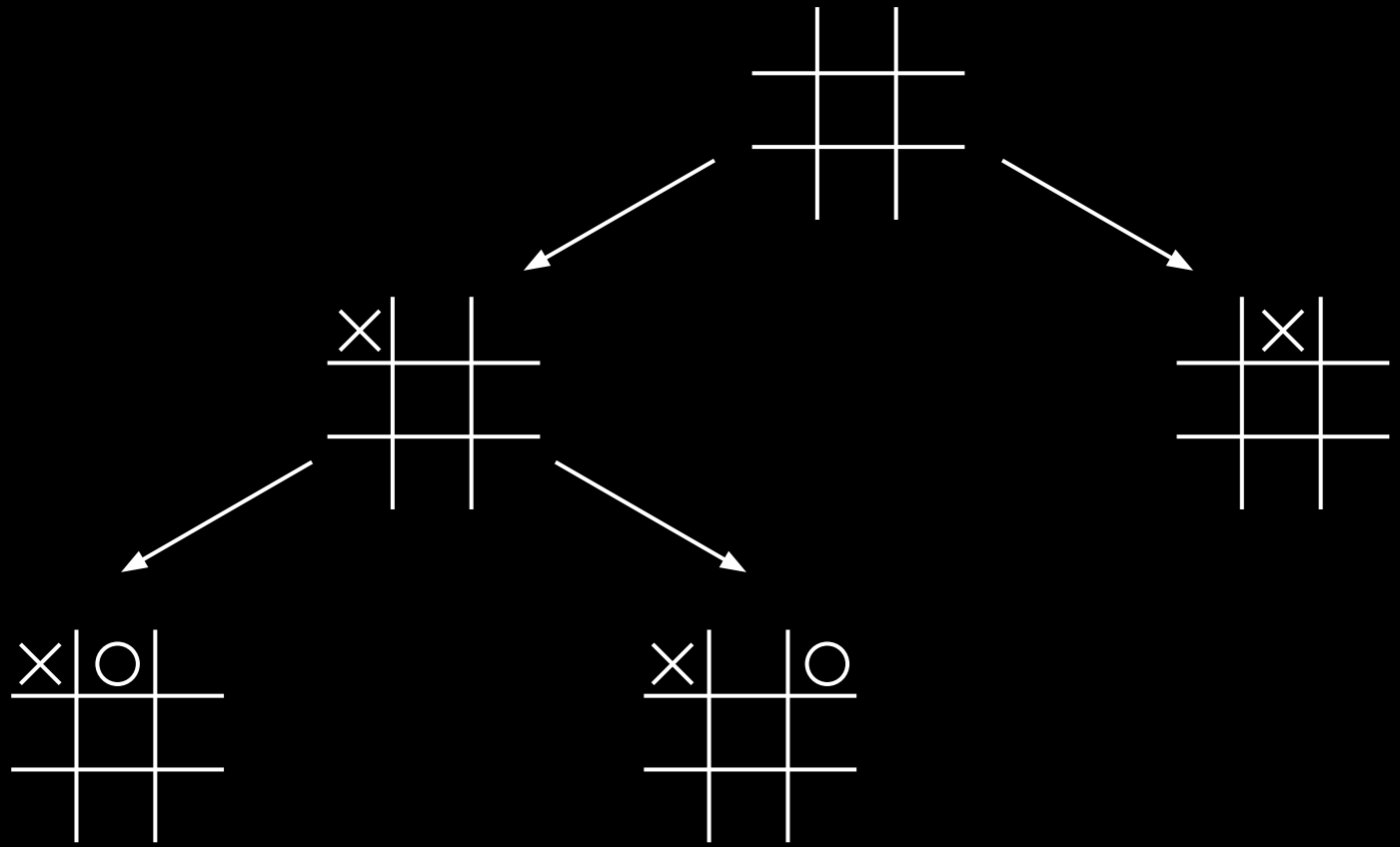
choose the move with the best value.

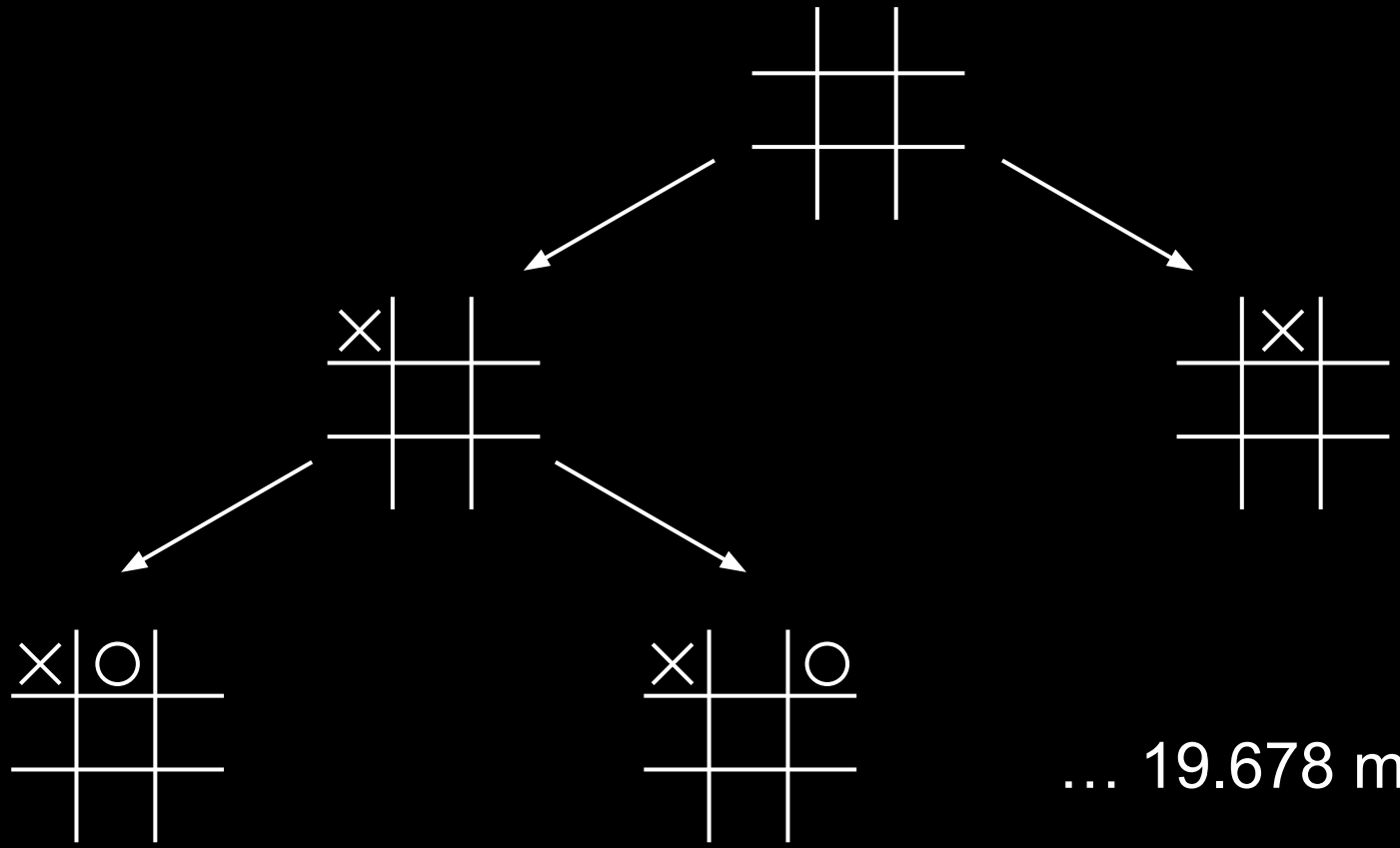






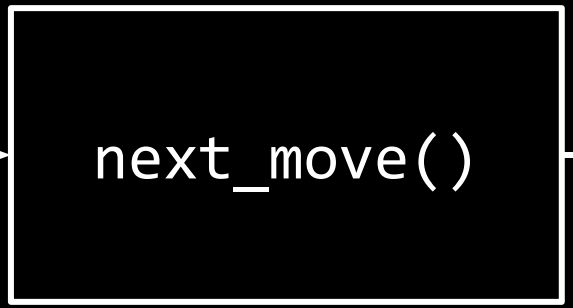






… 19.678 more

I. we can write rule-based programs to solve some problems

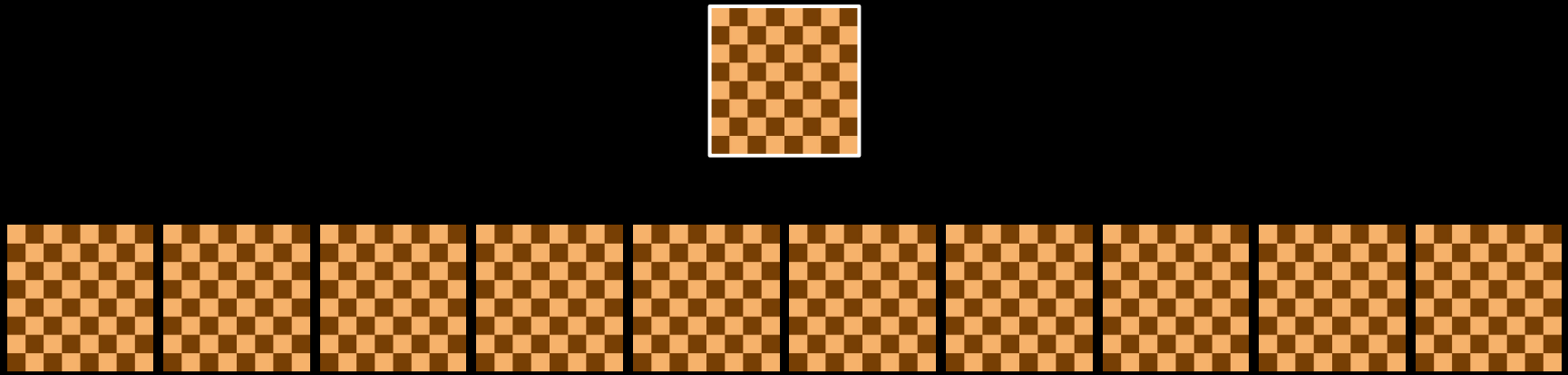


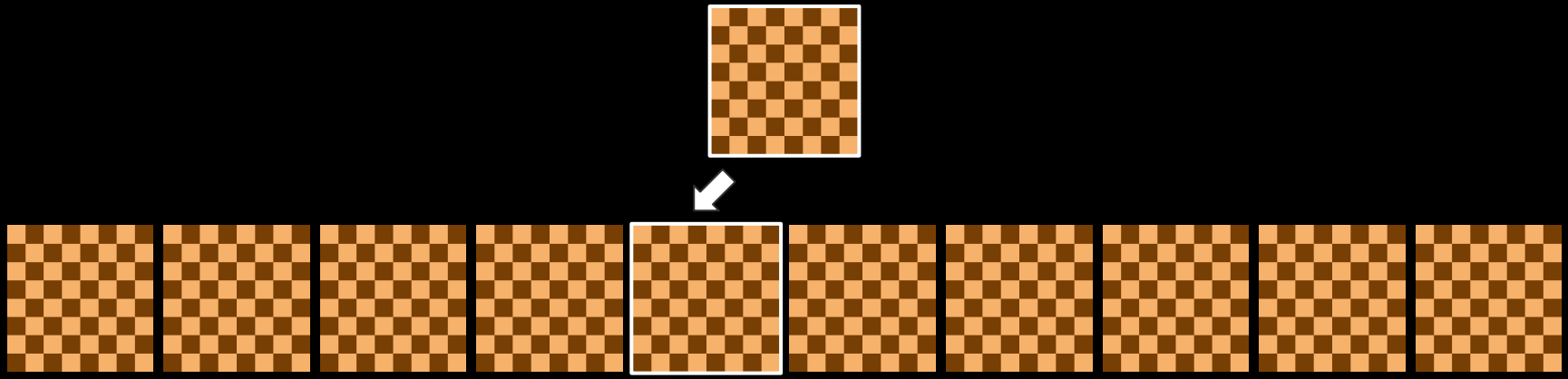
E2 → E4

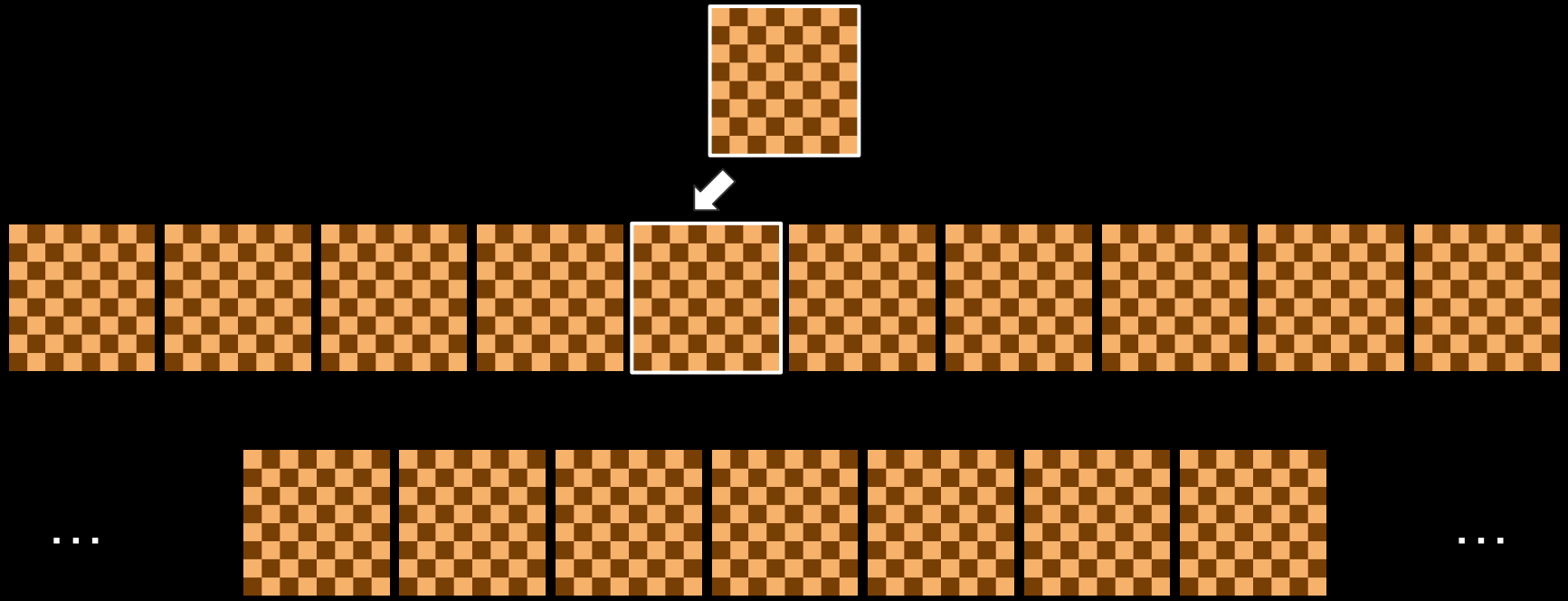
brute-force search ?

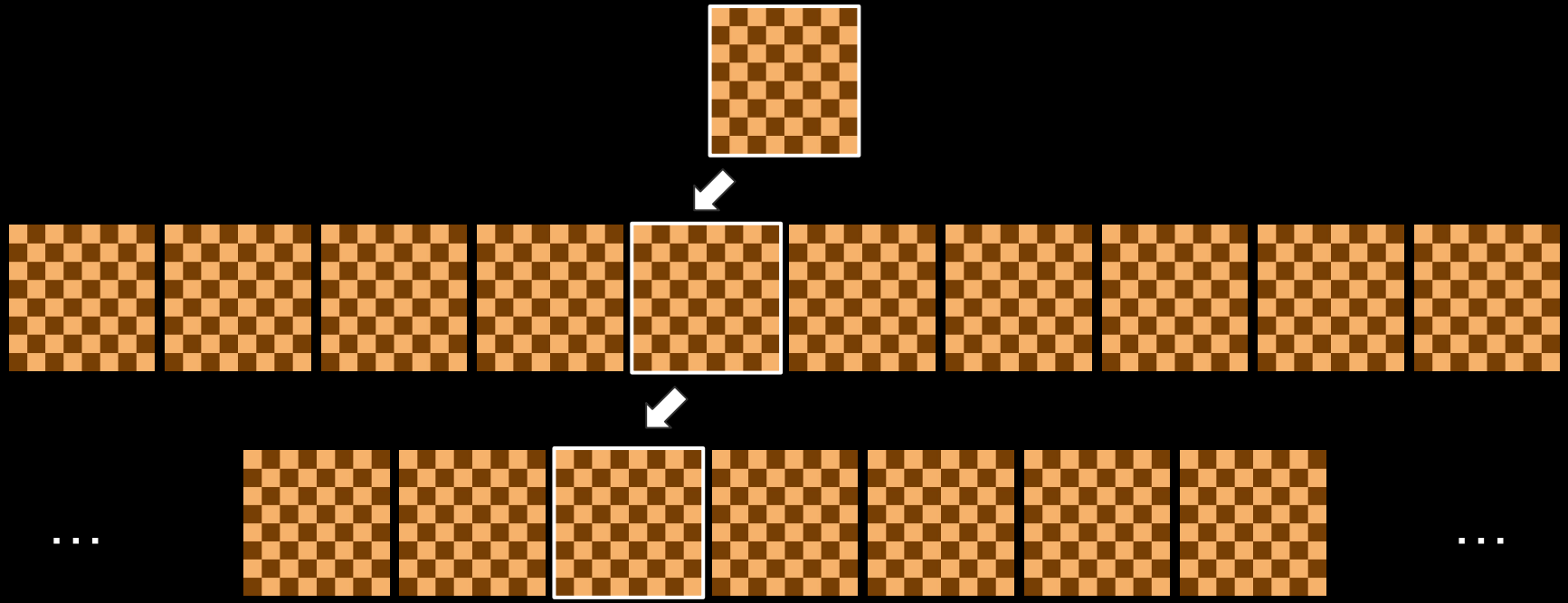
`next_move()`

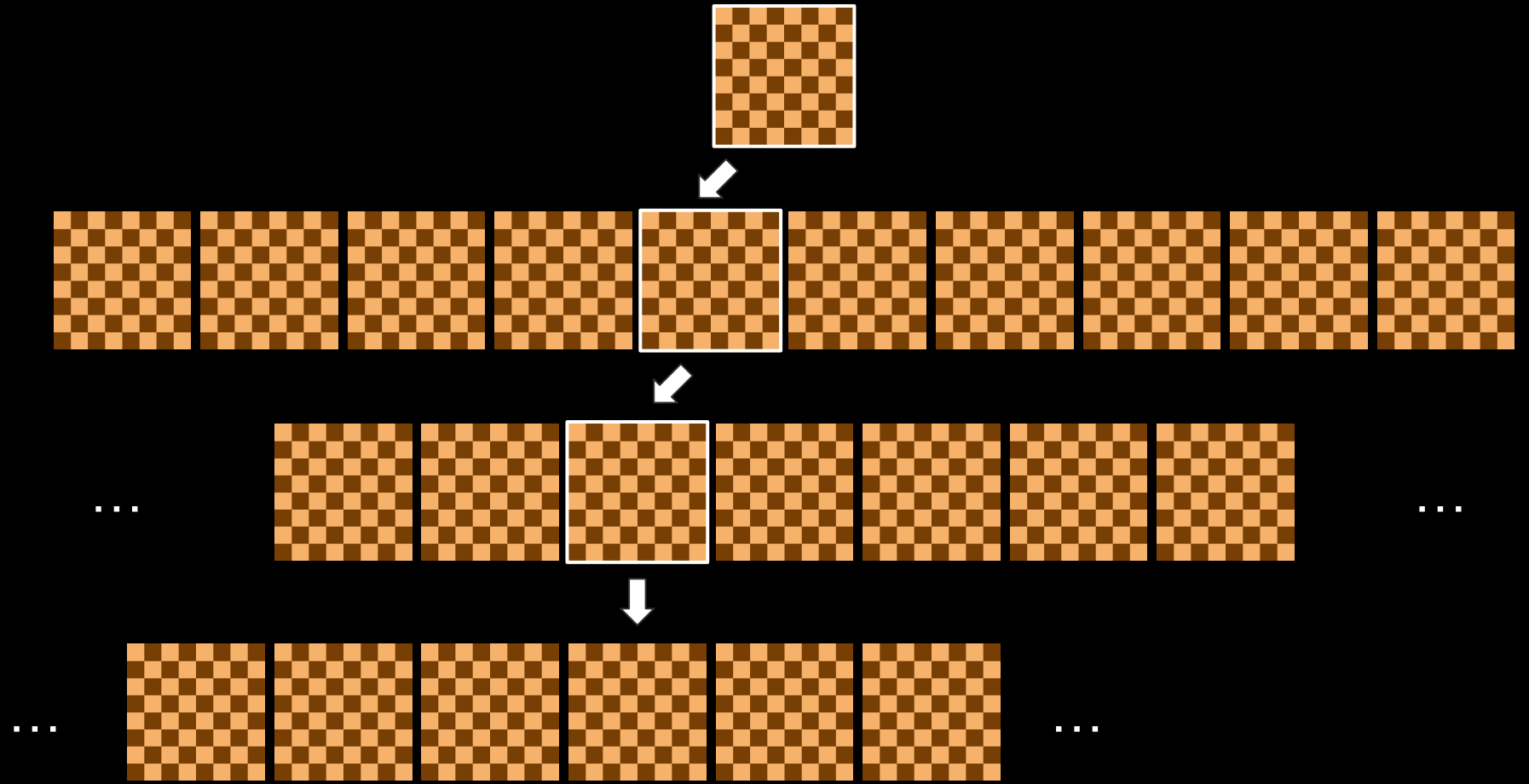
E2 → E4

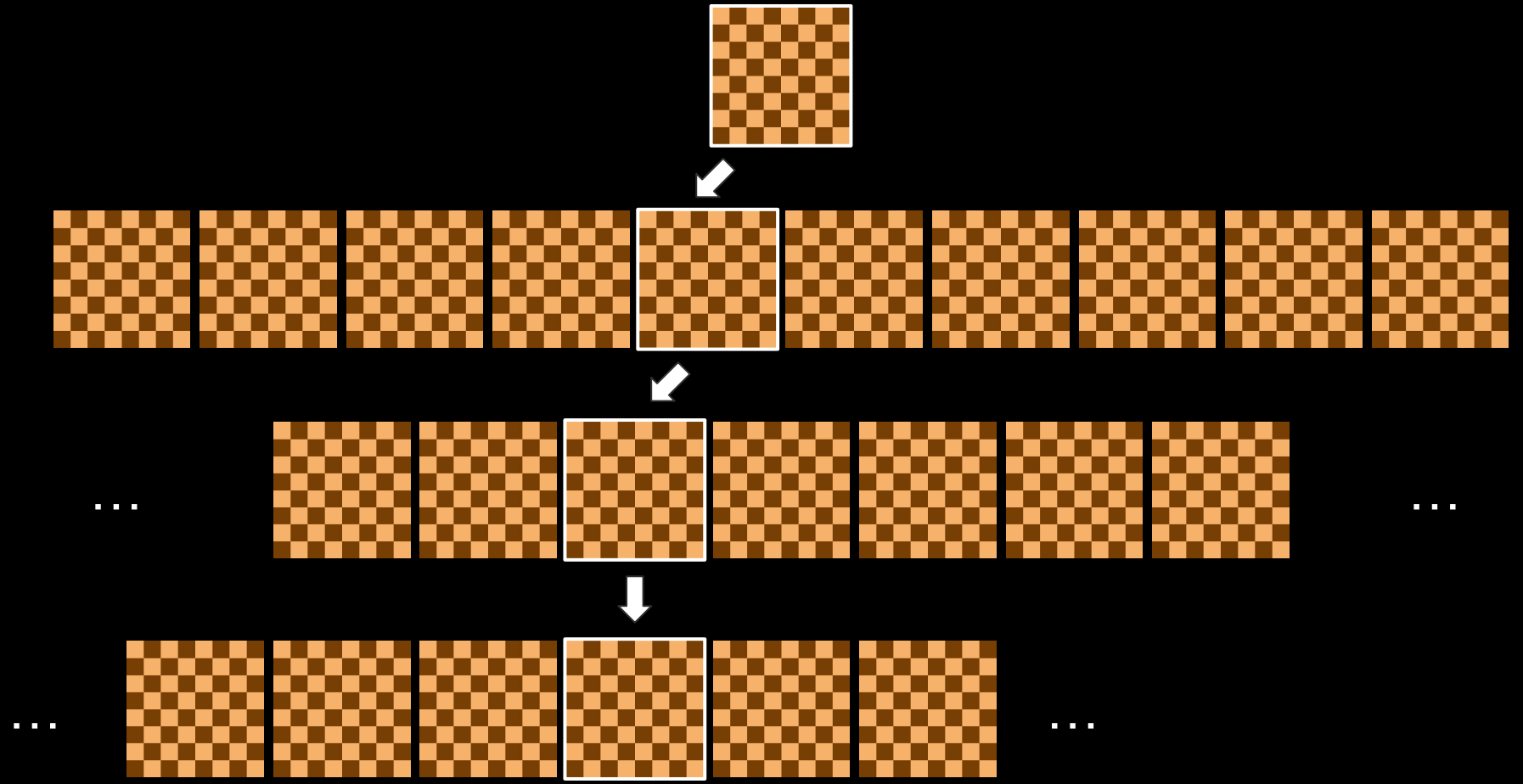












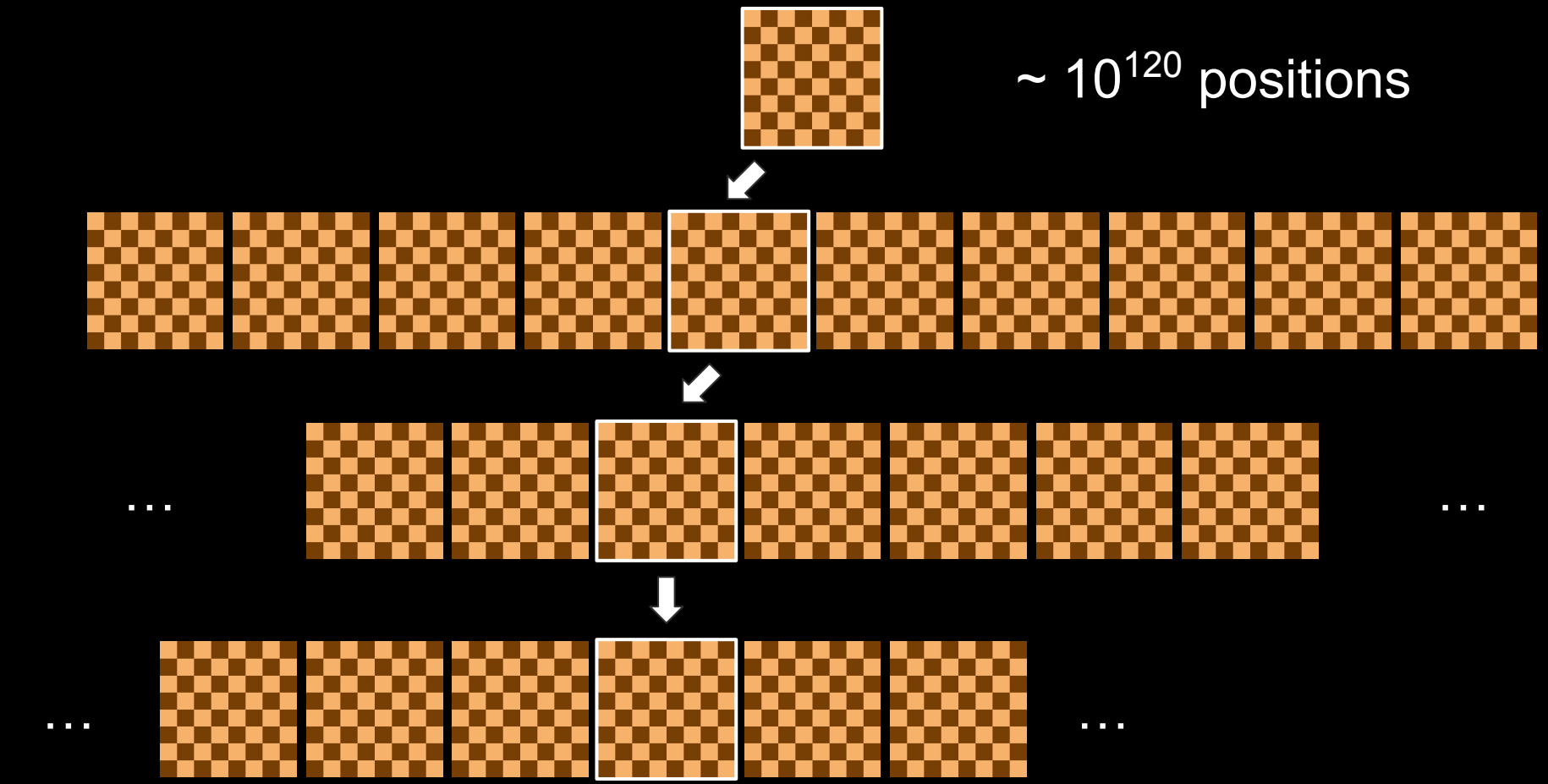

~ 10^{120} positions
...
...
...
...

Image source: IBM

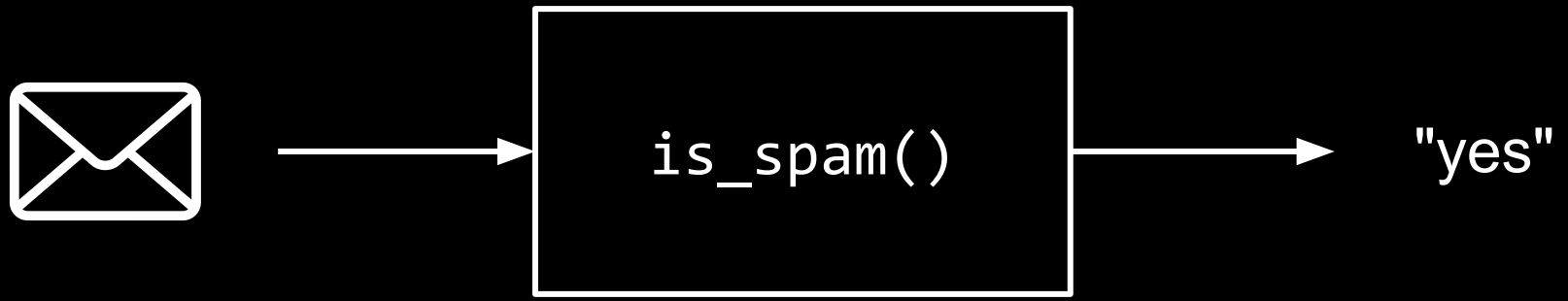
Image source: IBM

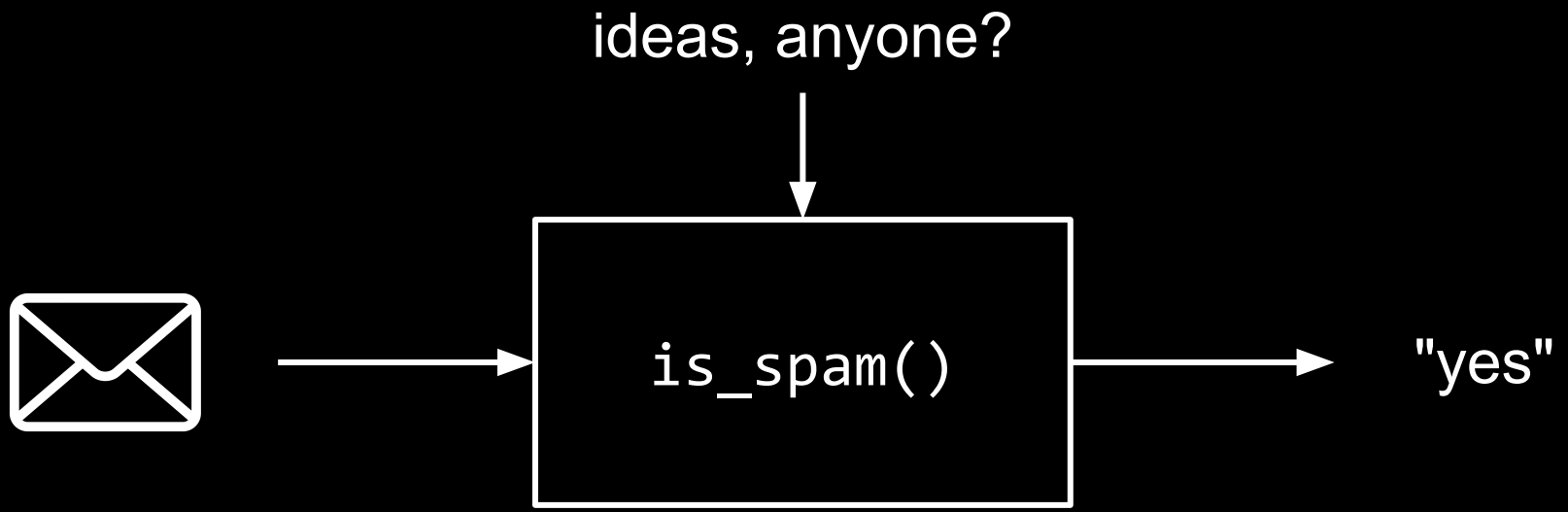
~ 200 billion positions per second

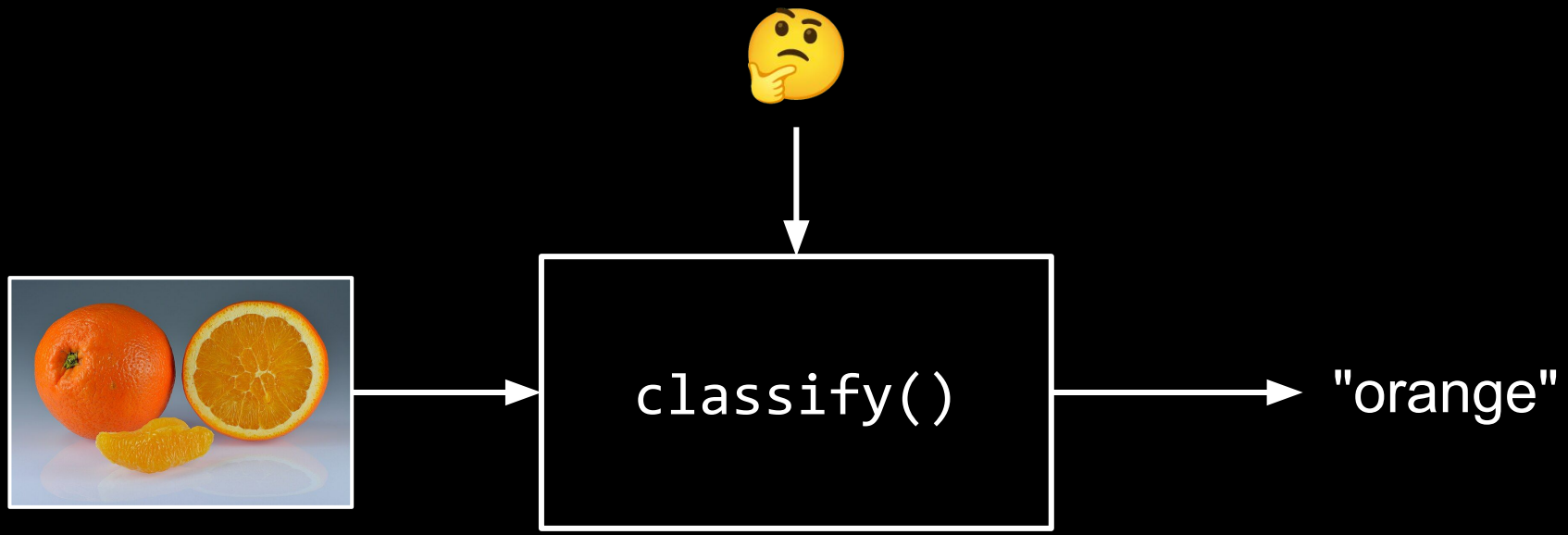
6-8 steps lookahead

I. we can write rule-based programs to solve some problems

II. some problems are too complex to be solved with rules alone


is_spam()
"yes"


ideas, anyone?
is_spam()
"yes"


classify()
"orange"

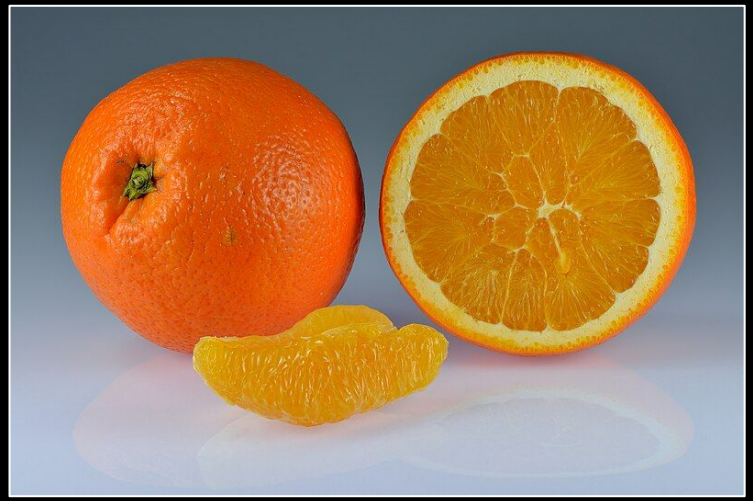


Image source: Wikimedia

rule-based attempt:

```
count all orange pixels.

if orange-ratio > 0.3:
   return "orange"
else
   return "no orange"
```

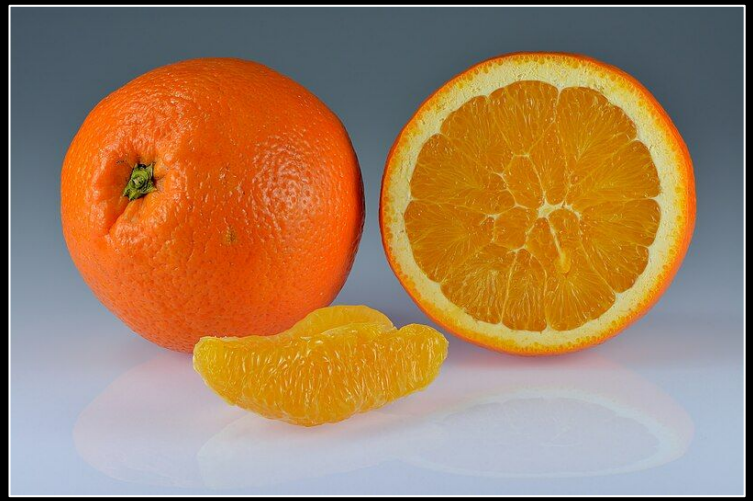


Image source: Wikimedia

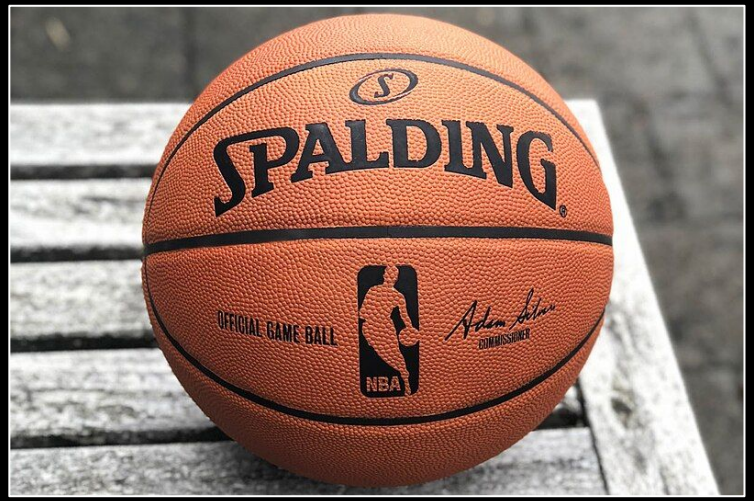
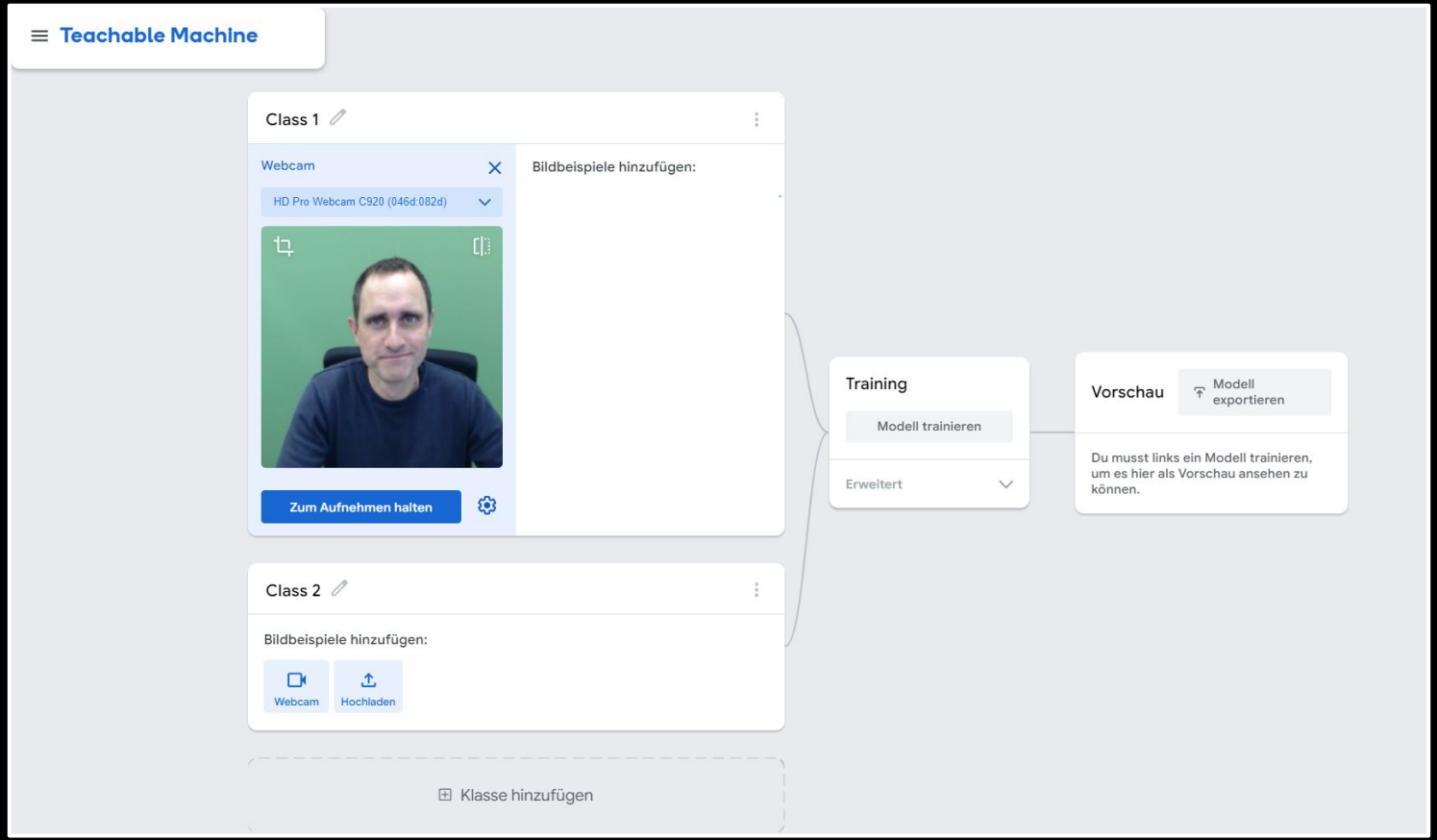
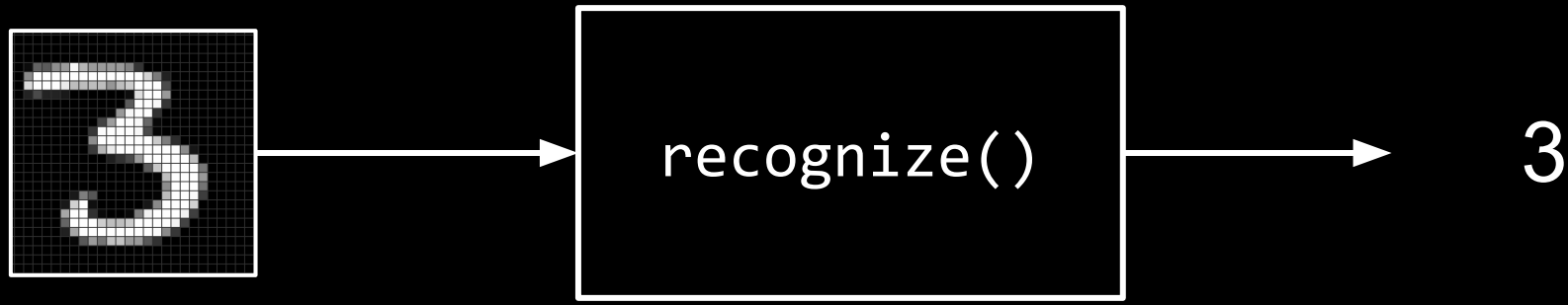
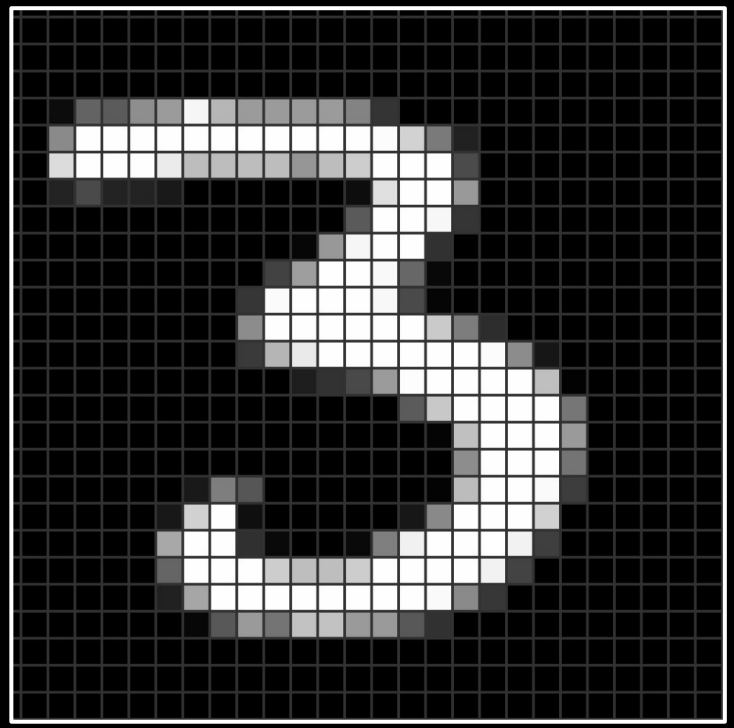


Image source: Wikimedia



https://teachablemachine.withgoogle.com/train/image


recognize()
3

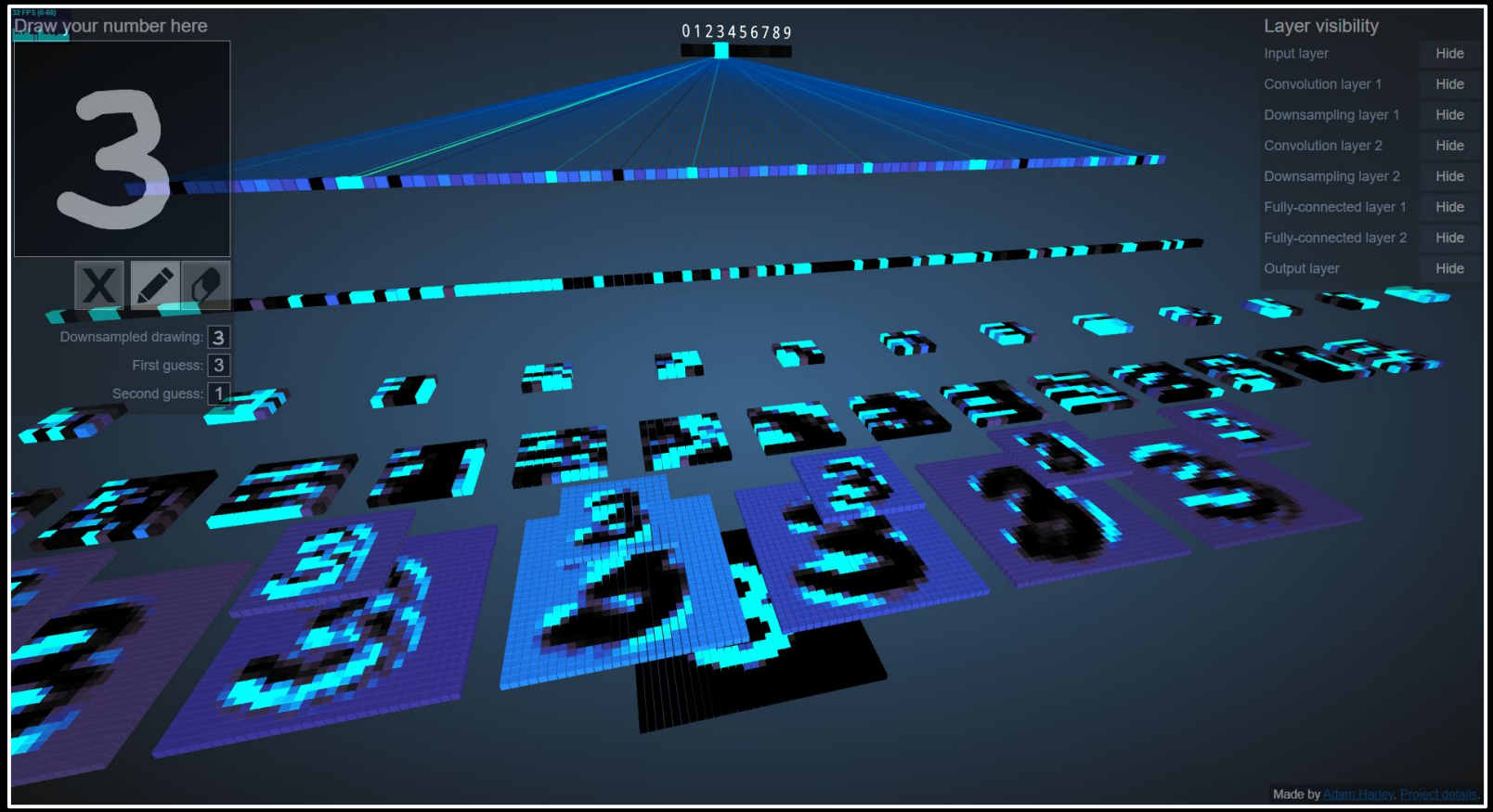


`rule-based attempt`:

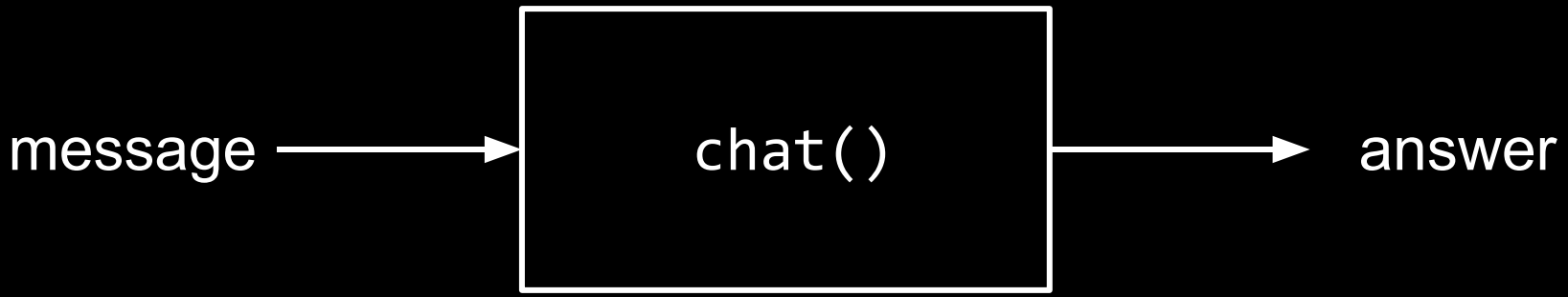
define a list of pixel positions for digit 3.

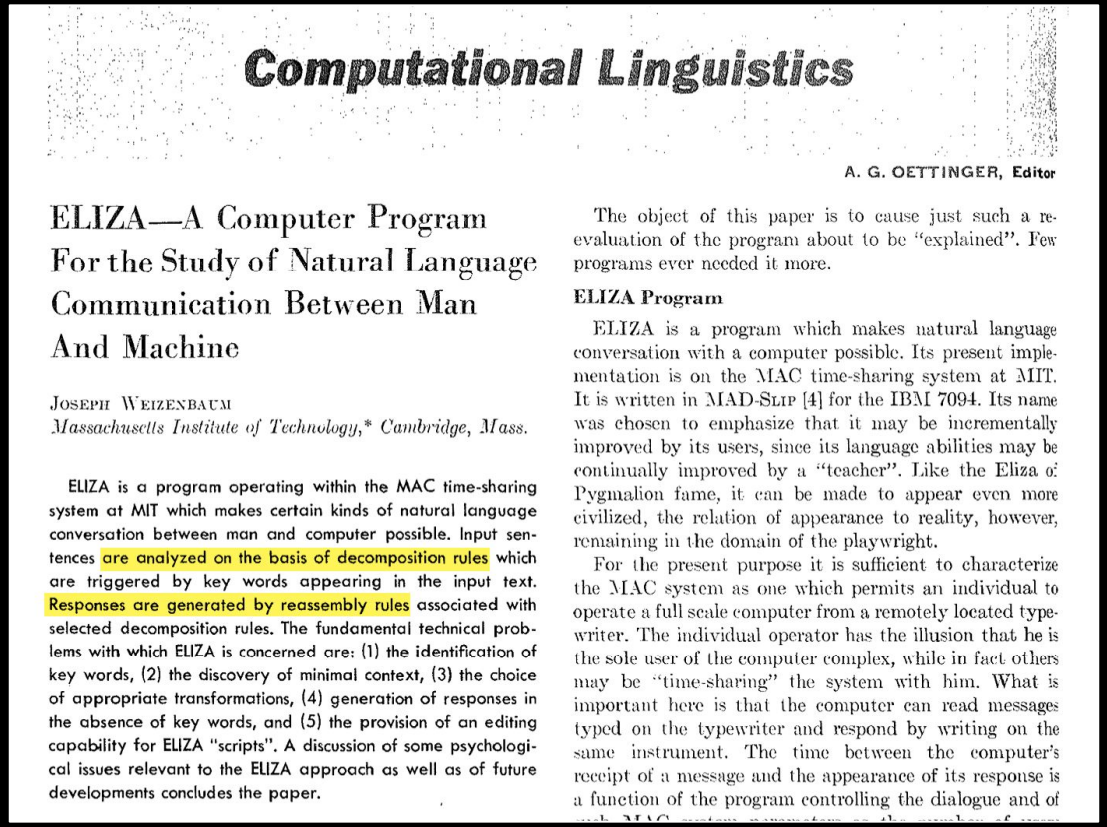
check each pixel and see if they match.

life is messy



https://adamharley.com/nn_vis/cnn/3d.html


message
chat()
answer



Computational Linguistics

A. G. OETTINGER, Editor

ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine

JOSEPH WEIZENBAUM
Massachusetts Institute of Technology, Cambridge, Mass.*

ELIZA is a program operating within the MAC time-sharing system at MIT which makes certain kinds of natural language conversation between man and computer possible. Input sentences are analyzed on the basis of decomposition rules which are triggered by key words appearing in the input text. Responses are generated by reassembly rules associated with selected decomposition rules. The fundamental technical problems with which ELIZA is concerned are: (1) the identification of key words, (2) the discovery of minimal context, (3) the choice of appropriate transformations, (4) generation of responses in the absence of key words, and (5) the provision of an editing capability for ELIZA "scripts". A discussion of some psychological issues relevant to the ELIZA approach as well as of future developments concludes the paper.

The object of this paper is to cause just such a re-evaluation of the program about to be "explained". Few programs ever needed it more.

ELIZA Program

ELIZA is a program which makes natural language conversation with a computer possible. Its present implementation is on the MAC time-sharing system at MIT. It is written in MAD-SLIP [4] for the IBM 7094. Its name was chosen to emphasize that it may be incrementally improved by its users, since its language abilities may be continually improved by a "teacher". Like the Eliza of Pygmalion fame, it can be made to appear even more civilized, the relation of appearance to reality, however, remaining in the domain of the playwright.

For the present purpose it is sufficient to characterize the MAC system as one which permits an individual to operate a full scale computer from a remotely located typewriter. The individual operator has the illusion that he is the sole user of the computer complex, while in fact others may be "time-sharing" the system with him. What is important here is that the computer can read messages typed on the typewriter and respond by writing on the same instrument. The time between the computer's receipt of a message and the appearance of its response is a function of the program controlling the dialogue and of

Weizenbaum, Joseph. "ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine." Communications of the ACM, vol. 9, no. 1, 1966, pp. 36–45, https://doi.org/10.1145/365153.365168.


prompt
ChatGPT
answer

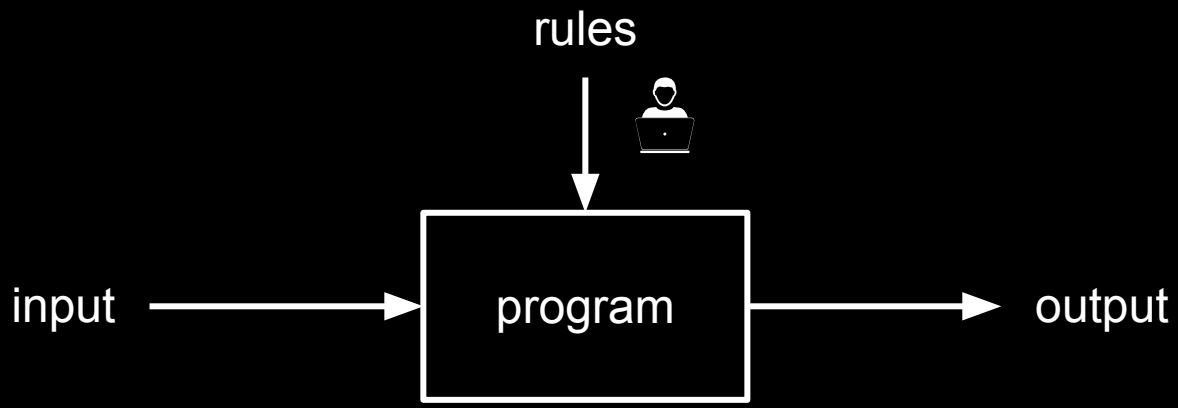
I. we can write rule-based programs to solve some problems

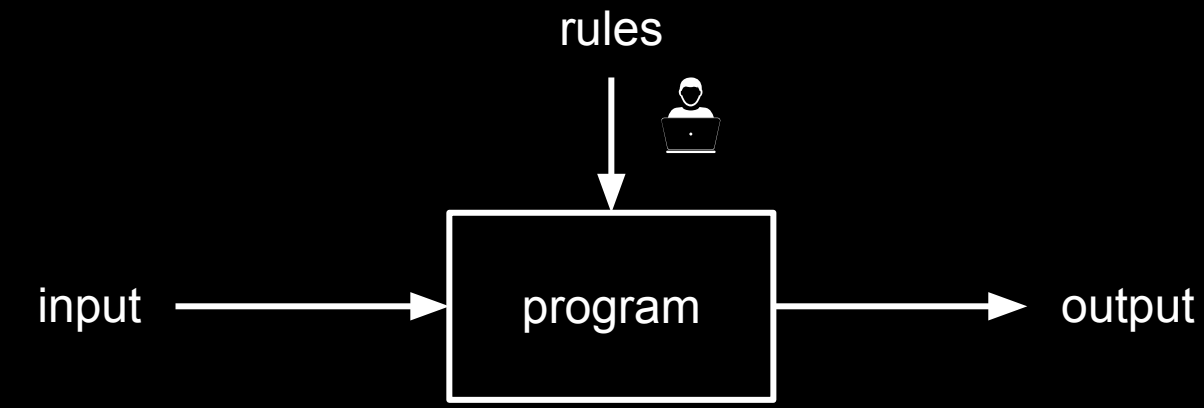
II. some problems are too complex to be solved with rules alone

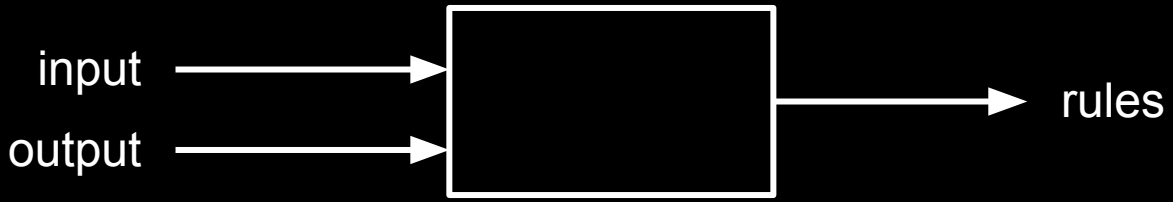
III. for some problems, we cannot define a suitable set of rules to solve it

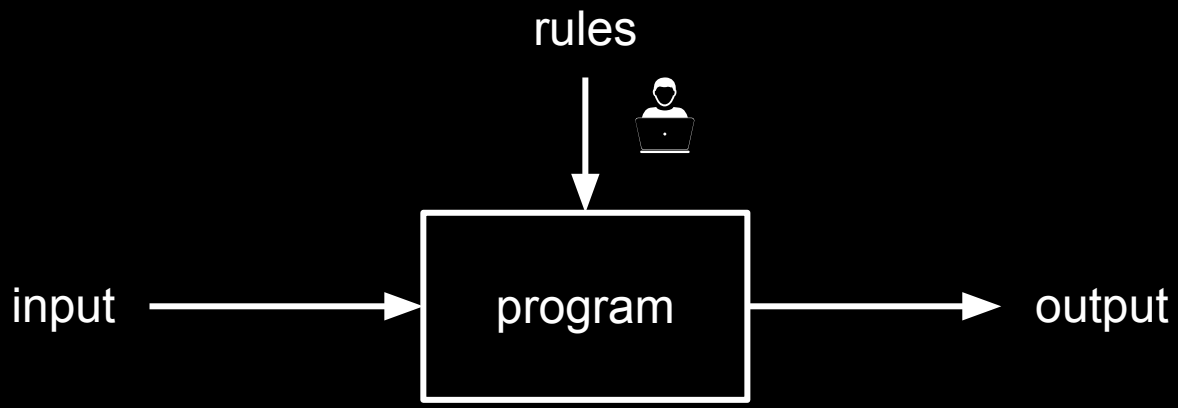
LEARNING

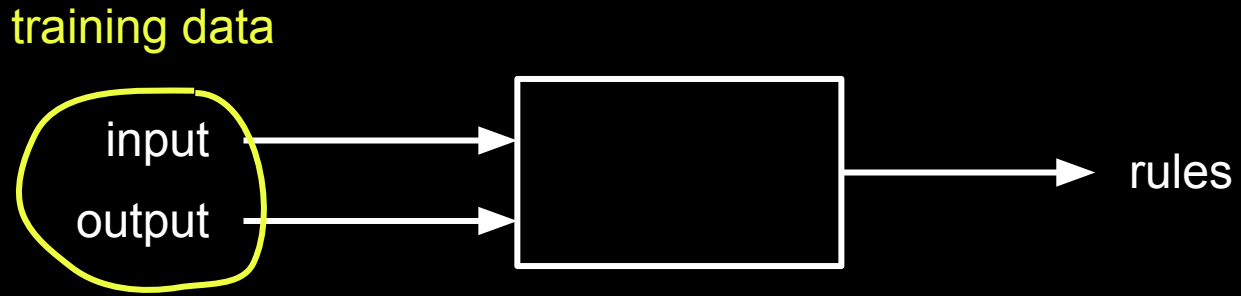

input
program
output

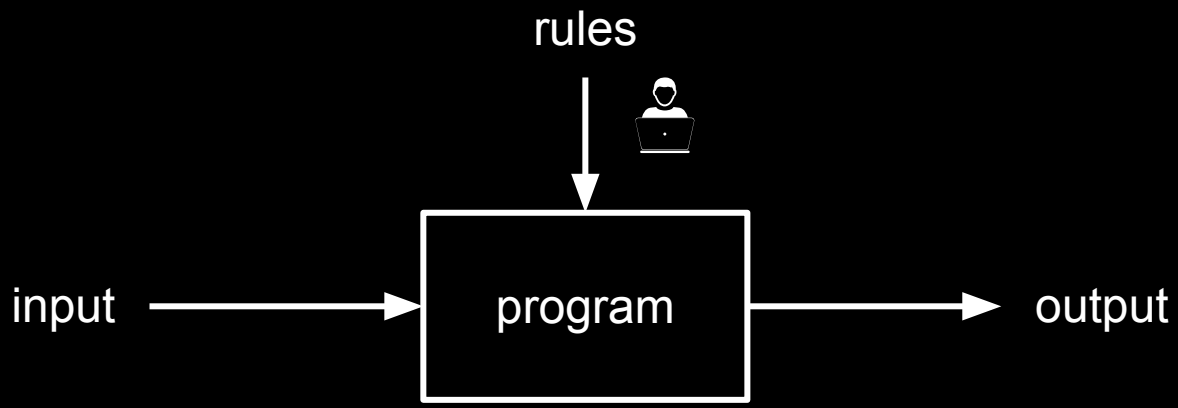

rules
input
program
output

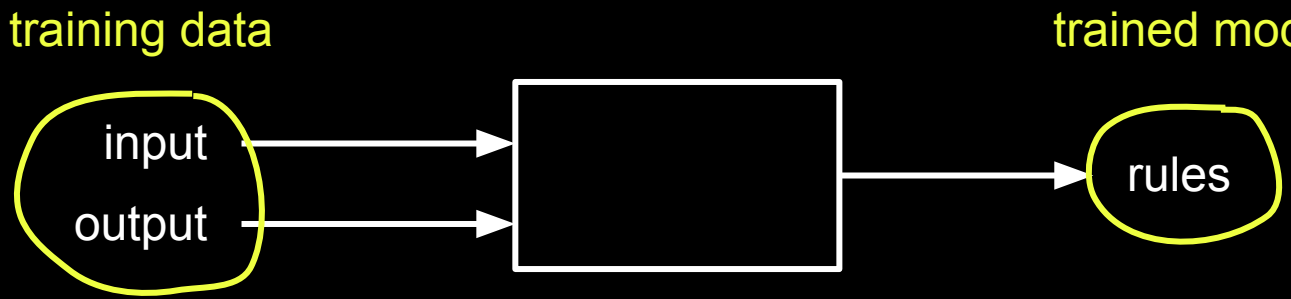

rules
input
program
output

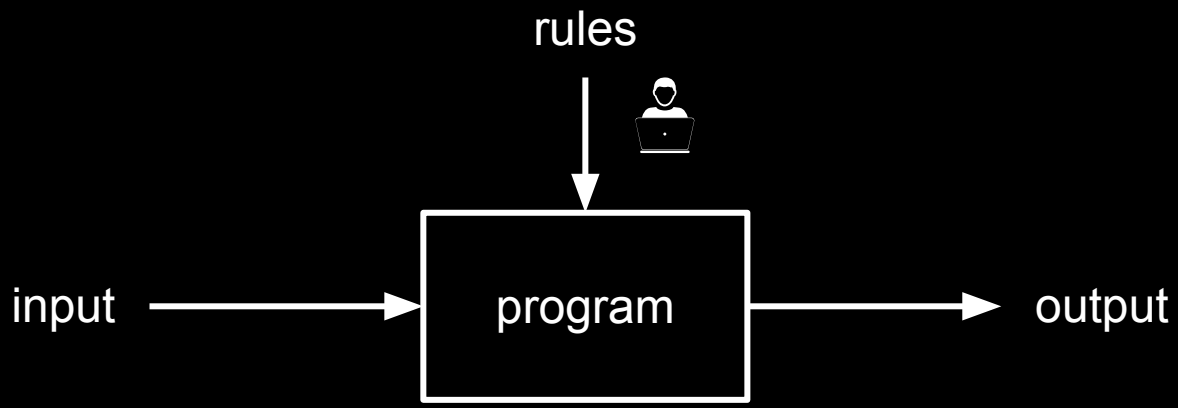

input
output
rules

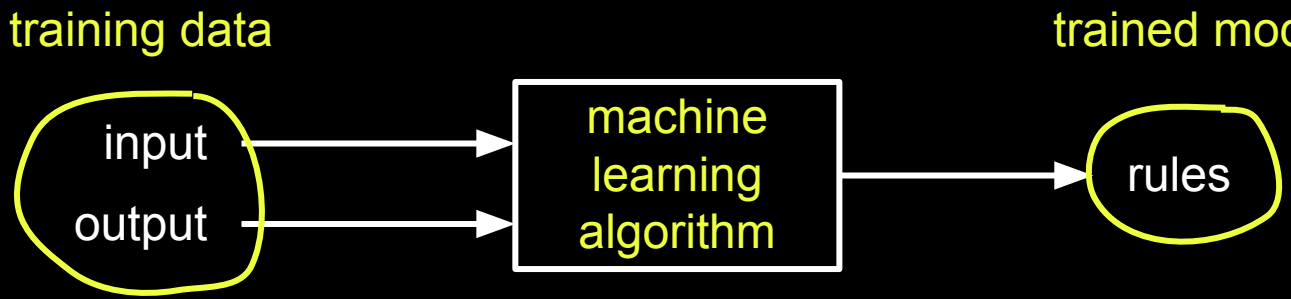

rules
input
program
output

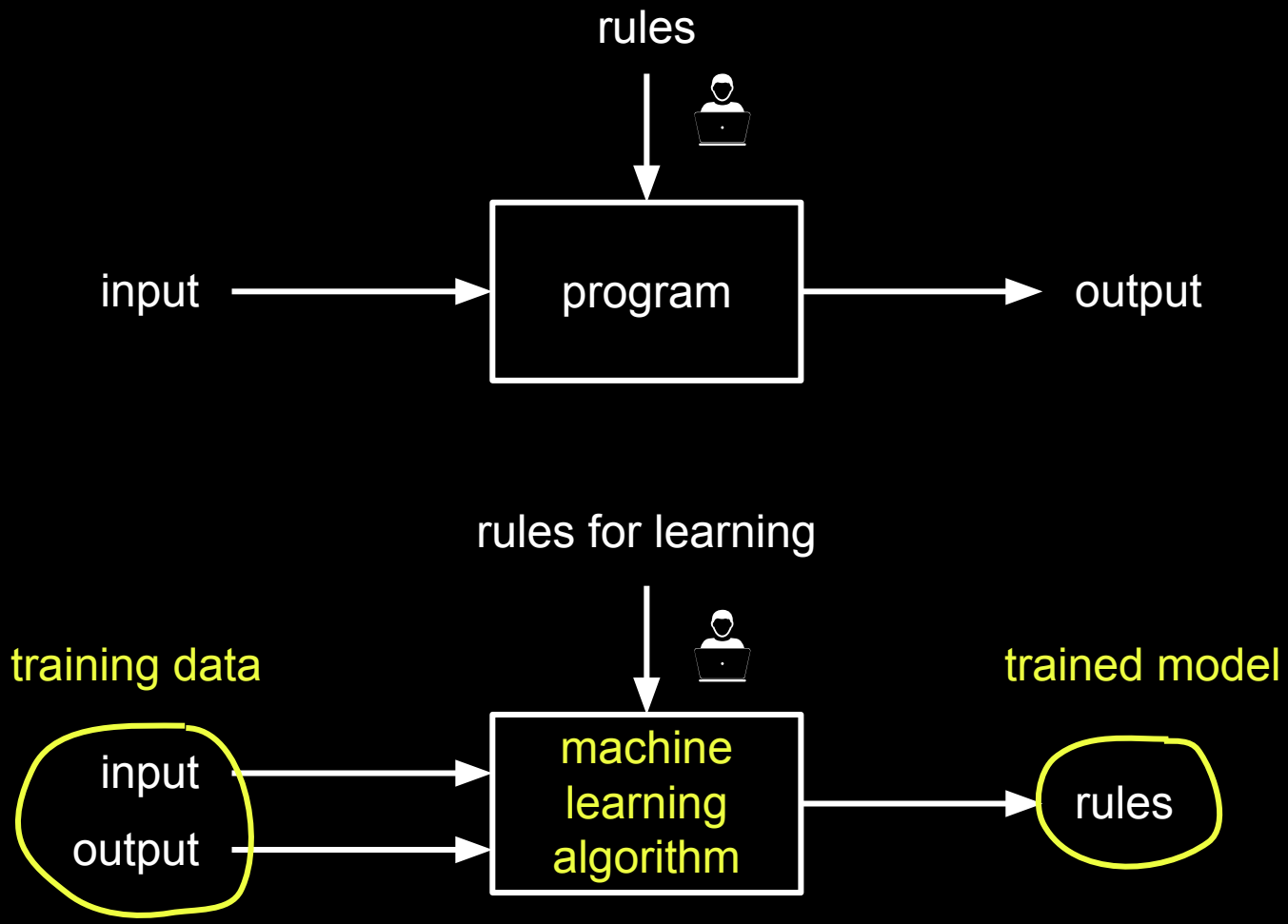

training data
input
output
rules

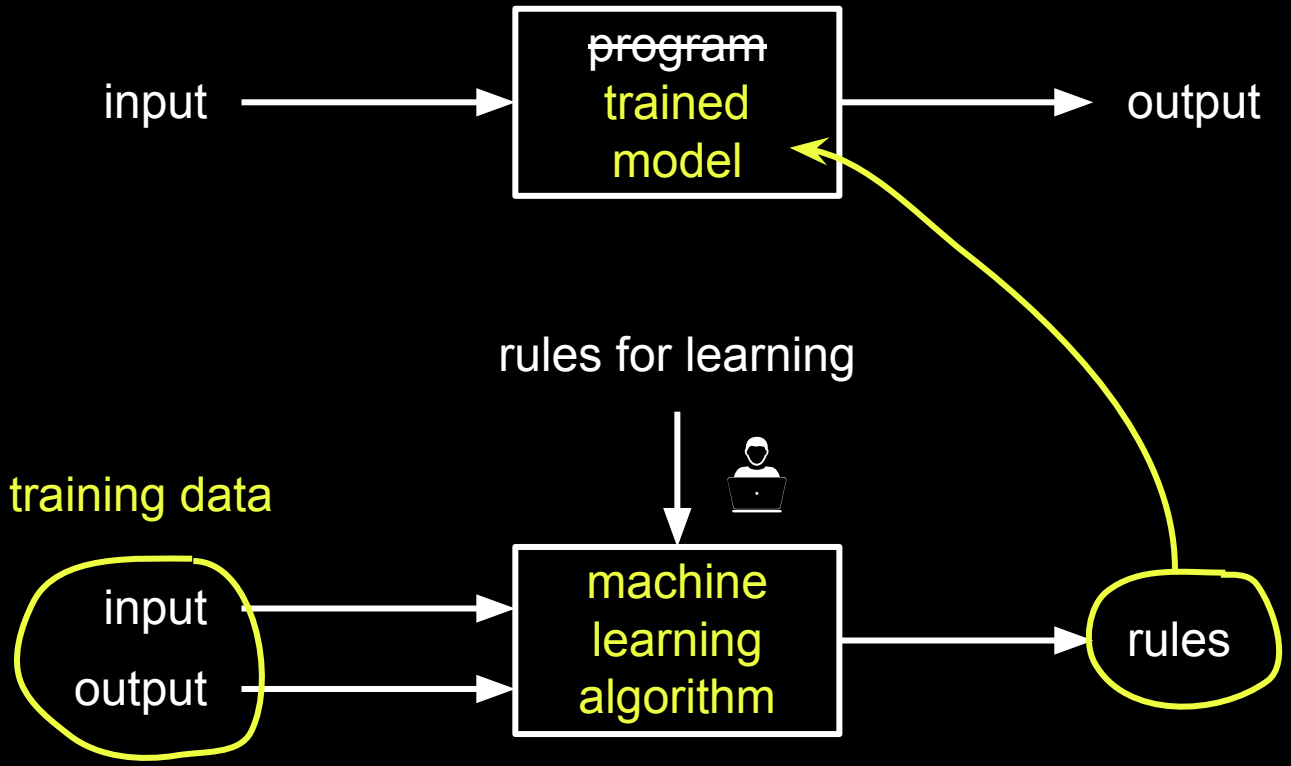

rules
input
program
output

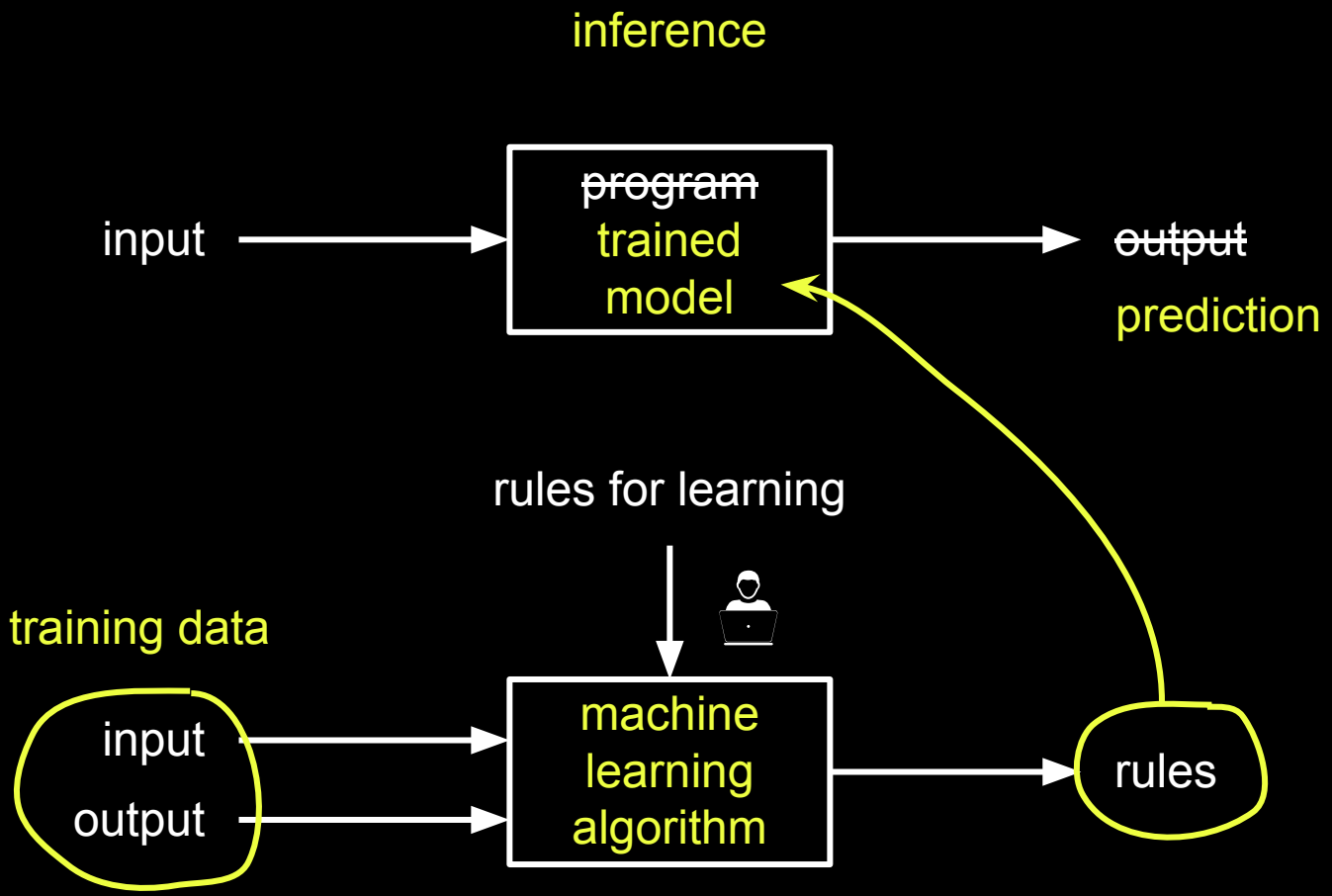

training data
trained model
input
output
rules

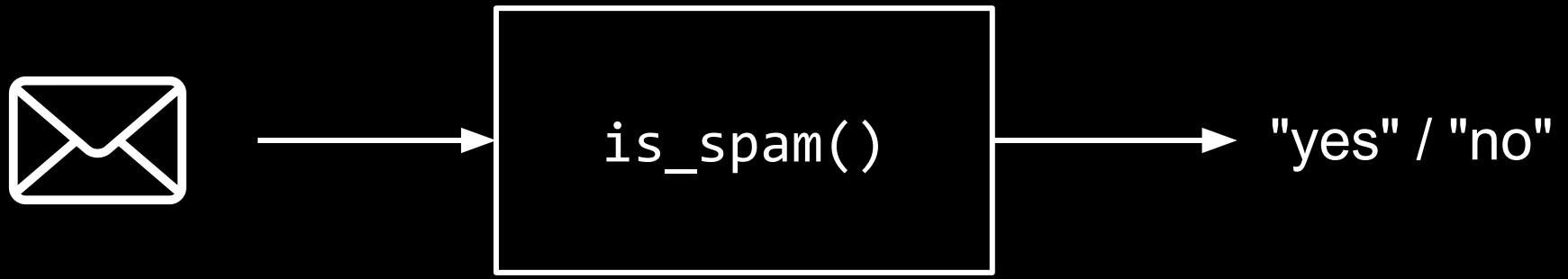

rules
input
program
output

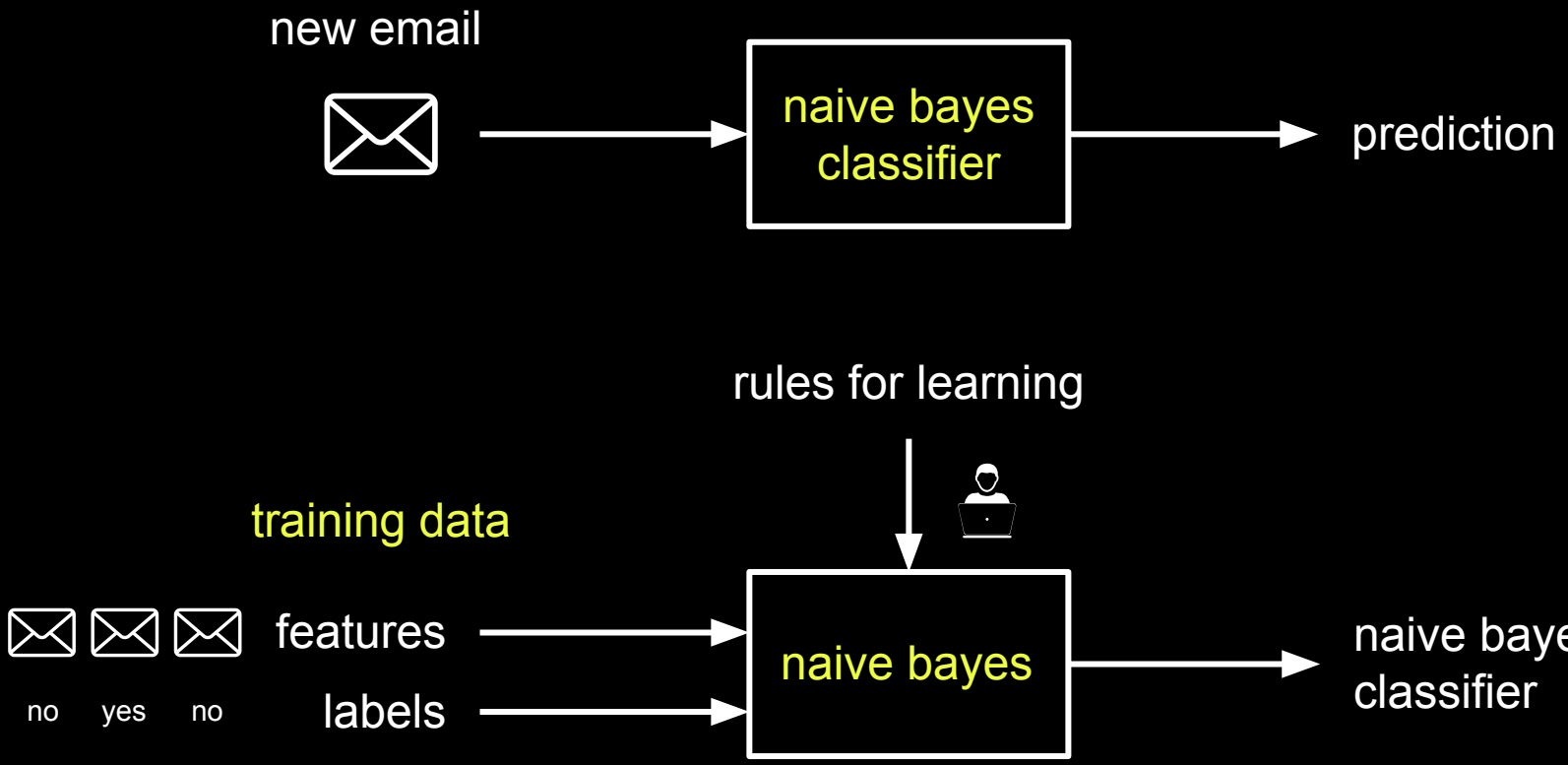

training data
input
output
machine
learning
algorithm
trained model
rules


rules
input
program
output
rules for learning
training data
trained model
input
output
machine
learning
algorithm
rules

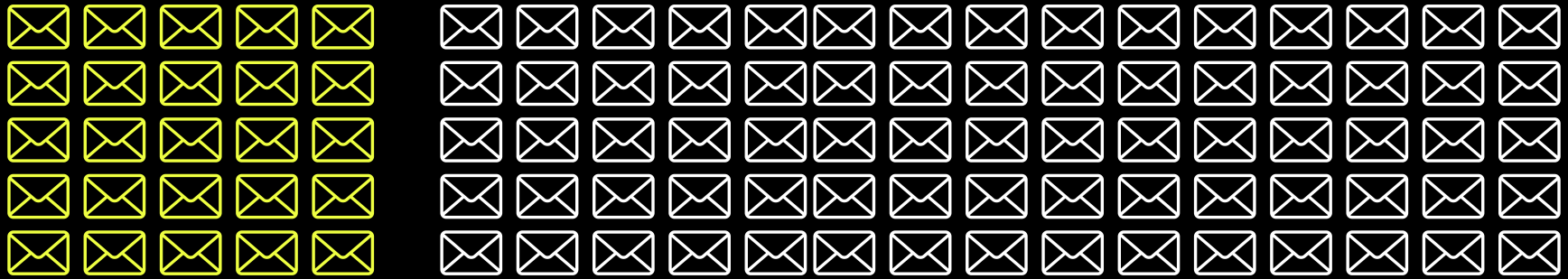

~~program~~
trained
model
input
output
rules for learning
training data
input
output
machine
learning
algorithm
rules


inference
input
~~program~~
trained
model
~~output~~
prediction
rules for learning
training data
input
output
machine
learning
algorithm
rules


is_spam()
"yes" / "no"


new email
naive bayes classifier
prediction
rules for learning
training data
features
labels
no
yes
no
naive bayes
naive bayes classifier

$$P(spam|words) = \frac{P(words|spam) \times P(spam)}{P(words)}$$



	spam		no spam	
total	25		75	
"buy"	20	20/25 = 0.8	5	5/75 ≈ 0.066
"cheap"	15	15/25 = 0.6	10	10/75 ≈ 0.133
"buy" & "cheap"	12	0.8 × 0.6 = 0.48	≈ 0.67	0.066 × 0.133 ≈ 0,0089

P(spam | buy & cheap) = 12 / (12 + 0.67) ≈ 0.947

I. we can write rule-based programs to solve some problems

II. some problems are too complex to be solved with rules alone

III. for some problems, we cannot define a suitable set of rules to solve it

IV. for some problems, we can have a computer program learn the rules from data (machine learning)

x_1	x_2	y
4	2	8
1	2	5
0	5	10
2	1	4
3	2	?

Source: Ananthaswamy, Anil. Why Machines Learn: The Elegant Math behind Modern AI. Dutton, 2024.

x_1	x_2	y
4	2	8
1	2	5
0	5	10
2	1	4
3	2	?

Source: Ananthaswamy, Anil. Why Machines Learn: The Elegant Math behind Modern AI. Dutton, 2024.

which function produces the observed pattern?

$$f(x_1, x_2)$$

Source: Ananthaswamy, Anil. Why Machines Learn: The Elegant Math behind Modern AI. Dutton, 2024.

in this case, it's easy to spot
a potential candidate

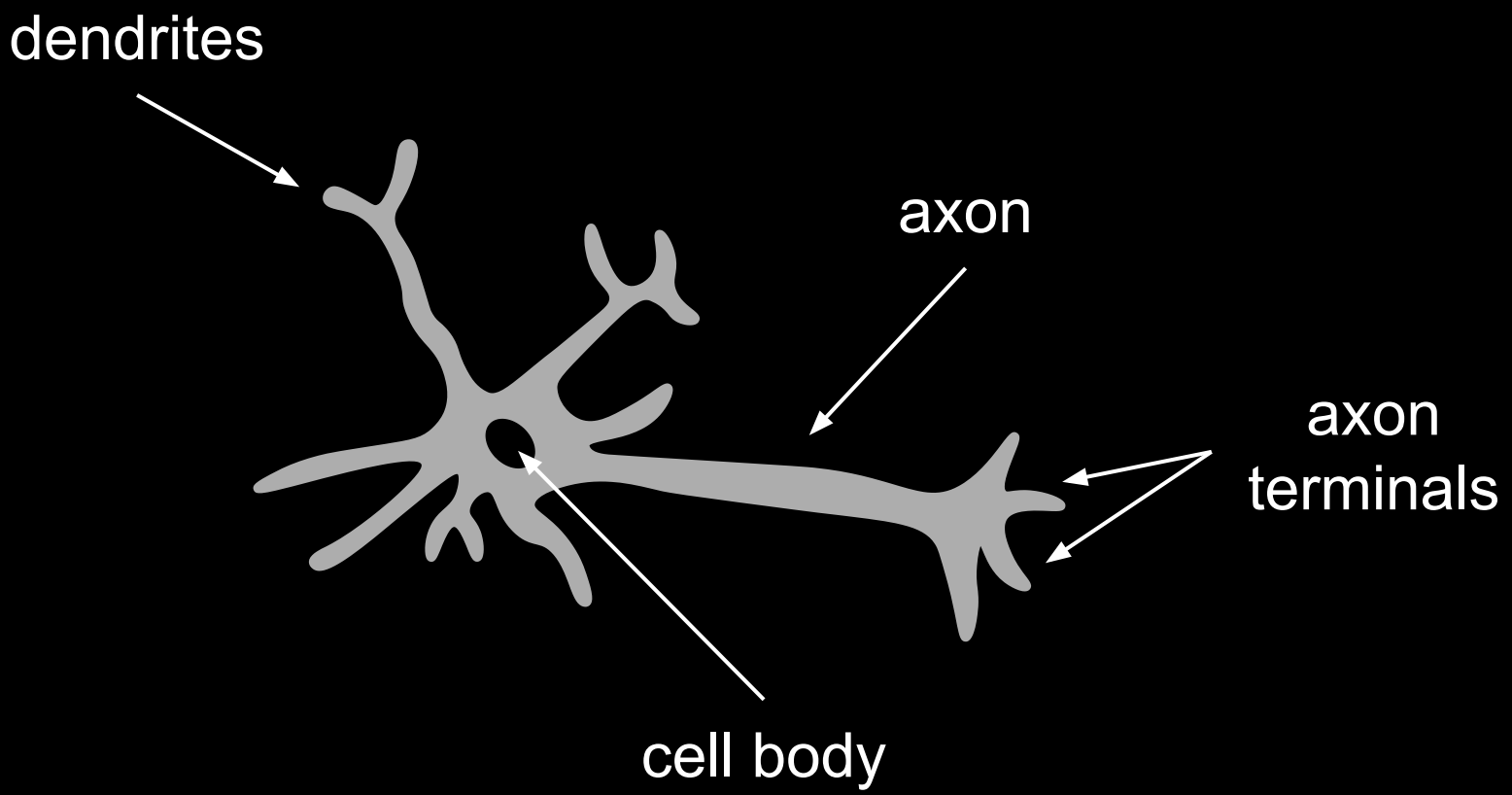
$$y = x_1 + 2x_2$$

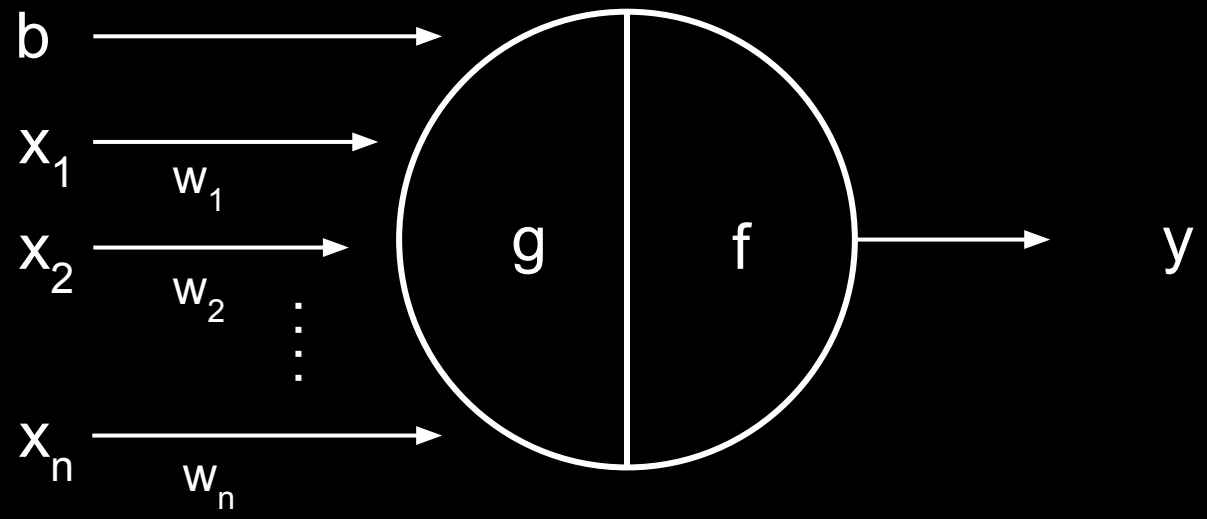
Source: Ananthaswamy, Anil. Why Machines Learn: The Elegant Math behind Modern AI. Dutton, 2024.

$$y = w_1 x_1 + w_2 x_2$$

what if it's not so easy?

Image generated with Flux Pro by Black Forest Labs


dendrites
axon
axon terminals
cell body



perceptron

multi-layered
perceptron

backpropagation

is my password safe?