Can Machines Think? *(Computer Science Meets Philosophy)*

Can Machines Think?

A Short Historical and Philosophical Discussion of Artificial Intelligence

Peter Millican

Gilbert Ryle Fellow and Professor of Philosophy, Hertford College, Oxford)

1

1. Alan Turing: mathematician, code-breaker, visionary

Mathematician

Turing's famous **1936 paper** introduced the theoretical "Turing machine" — an abstract model defining the limits of what is computable. These limits still apply to every algorithm run on every computer ever built.

Code-Breaker

At Bletchley Park, Turing led the team that cracked the Enigma cipher used by Nazi forces. His Bombe machine automated the decryption of German messages, shortening the war by an estimated two years.

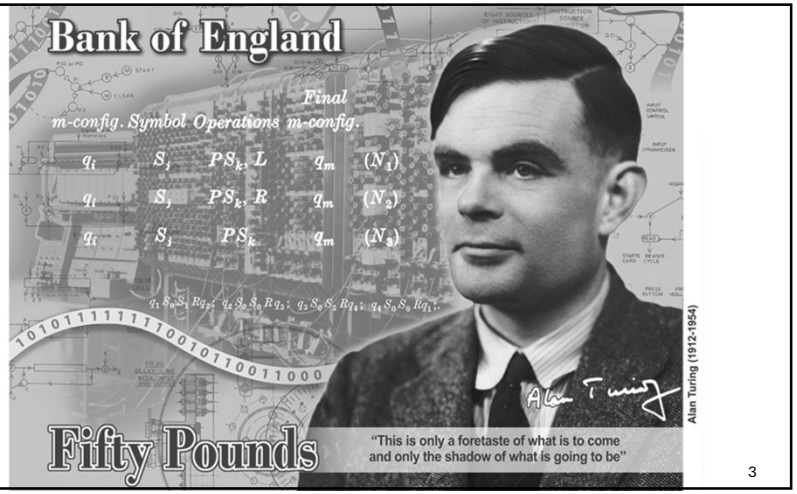
Visionary

His **1950 paper** "Computing Machinery and Intelligence" proposed the Turing Test for a thinking machine and also imagined learning machines. It provoked serious discussion of Artificial Intelligence long before the term existed.

The Turing Award – the highest honour in Computer Science – is named in his memory, and his face, together with a prophetic quotation and images of his work, is on the UK £50 note.

2

2



3

3

The Turing Machine (1936)

Here is a Turing Machine simulator, * running a program from §3 of his 1936 paper. It is generating an unending sequence of 0s and 1s, representing (in binary) the number:

[0.] 0 0 1 0 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 ...

(which is roughly 0.179184 in decimal).

The Turing Machine is driven by a table of very limited instructions (as shown in the image), and yet despite these strict limitations, it seems that by devising a suitable machine table, *any algorithm that can be performed by any computer (including a human computer), can be carried out by some Turing machine*. This **Church-Turing Thesis** is generally believed to be true, thanks to arguments in Turing's 1936 paper.

TAPE:

@ @ 0 0 1 0 1 x 1 x 0 1

CYCLES: 45 (press ESCAPE to return to menu)

MACHINE TABLE (Petzold p. 87, transcendental):

m-config.	symbol	operations	final m-config.
1 ('b')	Any	P@,R,P@,R,P0,R,R,P0,L,L	2 ('o')
2 ('o')	1	R,Px,L,L,L	2 ('o')
	0		3 ('q')
3 ('q')	None	P1,L	4 ('p')
	Any	R,R	3 ('q')
4 ('p')	x	P ,R	3 ('q')
	@	R	5 ('f')
	None	L,L	4 ('p')
5 ('f')	None	P0,L,L	2 ('o')
	Any	R,R	5 ('f')

4

4

Can Machines Think? *(Computer Science Meets Philosophy)*

The Turing Test (1950)



Interrogator
(here Alan Turing with a teletype that looks remarkably like an Enigma machine)



Typed Messages

Human respondent
(here Charles Babbage or Ada Lovelace)



Computer respondent

The computer pretends to be human, while the interrogator tries to work out which respondent is which. The computer "passes" the test if the Interrogator cannot reliably tell which is the computer and which is the human.

5

5

2. Classical AI: Three Examples

Classical AI (or "Good Old-Fashioned AI"), involves:

- representing knowledge and reasoning explicitly, using symbols and rules that the programmers or software engineers understand, and data structures defined and crafted by them, rather than learned from data.
- The direction of control is typically *top-down*, with abstract rules being applied to specific cases, rather than *bottom-up*.
- The algorithms and processes are *transparent*, in the sense that the chain of reasoning can in principle be followed step by step, to see exactly why the system reached a given conclusion.

We'll see three simple examples:

- A chatbot using *pattern-matching*;
- A Nim-playing program using *reinforcement learning*;
- A Tic-Tac-Toe program using *recursive minimax search*.

6

6

ELIZA (1966) – A Seductive Blind Alley

Men are all alike.

IN WHAT WAY?

They're always bugging us about something or other.

CAN YOU THINK OF A SPECIFIC EXAMPLE?

Well, my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE?

I need some help, that much seems certain.

WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP?

Perhaps I could learn to get along with my mother

TELL ME MORE ABOUT YOUR FAMILY.

My mother takes care of me.

WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?

My father.

…

WHAT ELSE COMES TO YOUR MIND WHEN YOU THINK ABOUT YOUR FATHER?

Bullies.

DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE?

7

7

Simple Patterns and Responses

• "… alike …"	"IN WHAT WAY?"
• "something or other"	"FOR EXAMPLE?"
• "My X"	"YOUR X?"
• "I need Y"	"WHAT IF YOU GOT Y?"
• "… my mother"	"… YOUR FAMILY …"
• "My mother …"	"WHO ELSE …?"
• <no matching pattern>	"DOES THAT HAVE ANYTHING TO DO WITH YOUR X?"

8

8

Classical Machine Learning – the game of Nim

We start with 20 coins laid out on a table:



You move first – remove 1, 2, or 3 coins. Suppose you remove 3:



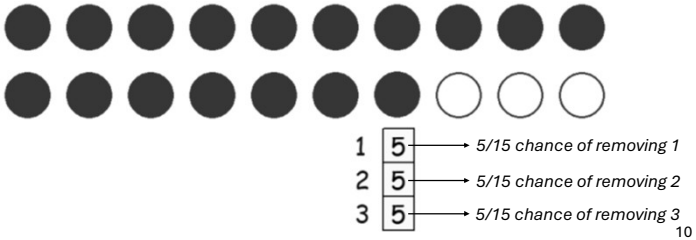
9

9

Now the computer can remove 1, 2, or 3 coins.

Then you and the computer continue to take turns doing the same. The winner is the player that picks up the LAST coin.

When the computer moves, it chooses its move using stored *probability weights*. For its first move, the weights are equal, with a 5/15 probability each of removing 1, 2 and 3 coins:



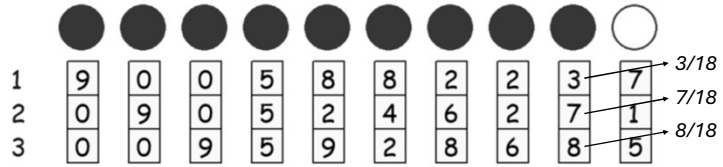
10

10

The computer keeps track of which moves have been played – it records your moves in GREEN, and its own moves in CYAN.

Then at the end of the game, the probability weights are adjusted so that the WINNER's moves are made MORE probable in future, while the LOSER's moves are made LESS probable. This is called **Reinforcement Learning** – a central technique in modern AI.

Here's how the weights for the last 10 coins might look soon, in a game where the computer is moving with 9 coins remaining: *



11

11

Introducing Recursion (with a simple triangle) *

```
def triangle(size):
    if (size>1):
        forward(size)
        right(120)
        forward(size)
        right(120)
        forward(size)
        right(120)

triangle(256)
```

This simple Python program consists of:

a) The triangle function, which draws a triangle of a given size.

b) A call to that function, which sets size to 256.

12

12

Making the Function Recursive

```
def triangle(size):
    if (size>1):
        forward(size)
        triangle(size/2)
        right(120)
        forward(size)
        triangle(size/2)
        right(120)
        forward(size)
        triangle(size/2)
        right(120)
```

We have added three *recursive calls* to the triangle function

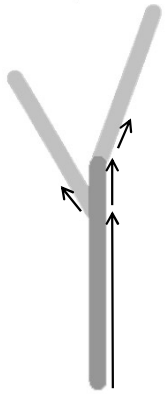
13

13

More Recursion (with a line) *

Now we have a function which draws a green line in two steps (as shown by the red arrow), stopping en route to make two temporary turns (the yellow arrows).

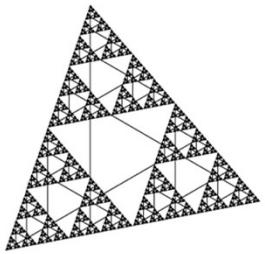
While turning, the function might call itself, in which case it will draw further lines as shown (in pink). The pattern gets more complex with each added level.

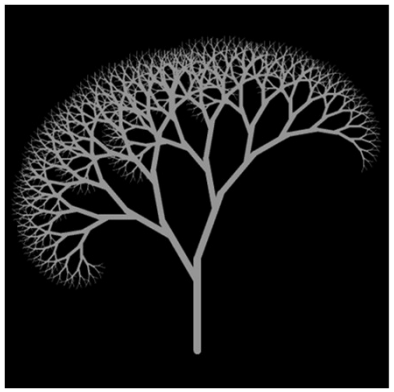




14

14



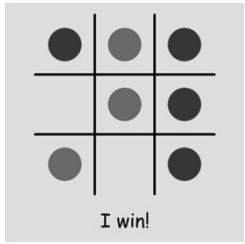


15

15

Tic-Tac-Toe (Noughts and Crosses) *

We'll work out a short recursive algorithm for playing error-free Tic-Tac-Toe: an algorithm that never loses, and always takes advantage of its opponent's errors.



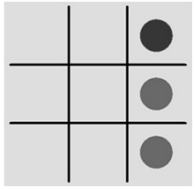
16

16

Can Machines Think? *(Computer Science Meets Philosophy)*

Playing infallibly requires a reliable method for assessing the "value" of any position (+1=winning; 0=drawing; -1=losing).

Consider this position in a game between Rosy (Red) and Bill (Blue), with Rosy to move.



There are 6 spaces left – let's say it's a "6-position".

After Rosy makes her move, her opponent, Bill, will be faced with a 5-position in which to make his reply.

Suppose Rosy knew a method for calculating *the value to Bill* of any 5-position that he may be faced with ...

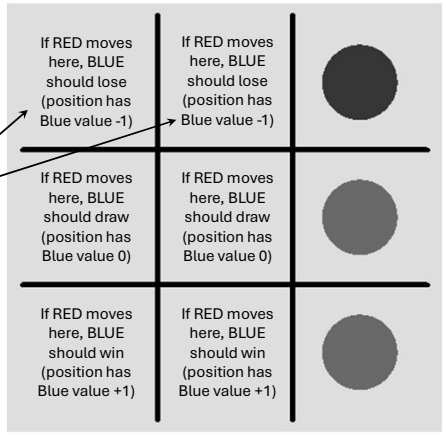
Puzzle: How could Rosy use this method to work out her own best move in this 6-position?
And could this also tell Rosy *the value to her* of this 6-position?

17

17

What Rosy could know if she had such a method ...

Best moves for RED (Rosy), since these yield the 5-positions that have the lowest value for BLUE (Bill)



18

18

"Minimaxing"

Rosy winning is the same as Bill losing (and vice-versa).

Hence the value of a position to Rosy is the inverse of its value to Bill (e.g. -1 as opposed to +1).

So Rosy tries to find moves which *maximise* the value to her, by *minimising* the value to Blue.

Thus Rosy, faced with a 6-position, *should choose the move which produces the 5-position which is **worst** – or equal-worst – for Bill*. (And that automatically tells her the value of the position to her.)

"OK, *but this was assuming Rosy has some method for calculating the value to Bill of any 5-position that he might be faced with*."

Puzzle: Does the reasoning above give the key to this method?

19

19

The Magic of Recursion

We have seen how to evaluate a 6-position if we can evaluate a 5-position,

... and in exactly the same way we can evaluate a 5-position if we can evaluate a 4-position,

... and we can evaluate a 4-position if we can evaluate a 3-position,

... and we can evaluate a 3-position if we can evaluate a 2-position,

... and we can evaluate a 2-position if we can evaluate a 1-position,

... and we can evaluate a 1-position if we can evaluate a 0-position!

The only additional requirement here is to be able to recognise a FINAL position as winning for Red, winning for Blue, or drawn.

20

20

So to evaluate a position with *A*'s turn to move:

- Has *A* already lost (because the opponent has a line of three)?
  If so, its *A*-value is -1.
- Is the position full up (with all nine spaces filled)?
  If so, its *A*-value is 0.
- If neither of these, then construct in turn all of the positions (with *B* to move) that can arise from the current position (i.e. try out in turn all of *A*'s possible moves in the current position) ...

  Evaluate each of these new positions *from b's point of view* (resulting in evaluations B_1, B_2, and B_3, say),
  then assign, as *A*-value of the *current* position,
  the *inverse* of the *lowest* of B_1, B_2, and B_3.

A should, of course, choose the move with the best *A*-evaluation.

21

21

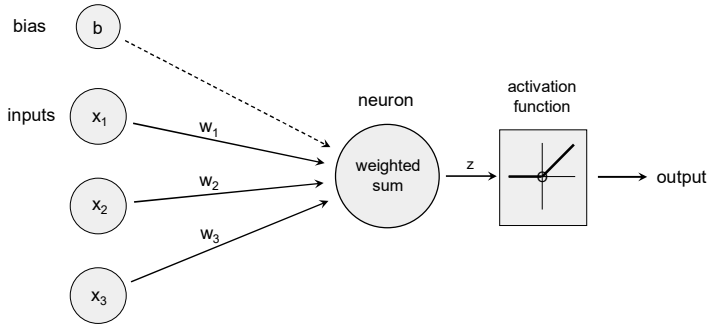
3. The Age of Deep Learning

- Three enablers of the deep learning breakthrough: internet-scale data, powerful graphics processing units (GPUs), and improved algorithms.
- 2012: AlexNet proves the power of deep networks by winning an image-classification competition by a huge margin over previous systems.
- 2015: DeepMind's Deep Q-Network learns to play a range of Atari video games purely from the raw pixel input and the score.
- 2016: DeepMind's AlphaGo defeats Go world champion Lee Sedol. A year later, AlphaZero learns multiple strategy games by self-teaching.
- 2017: "Attention Is All You Need" — the Transformer architecture is born.
- 2022: ChatGPT reaches 100 million users within two months.

22

22

The Perceptron (artificial neuron)

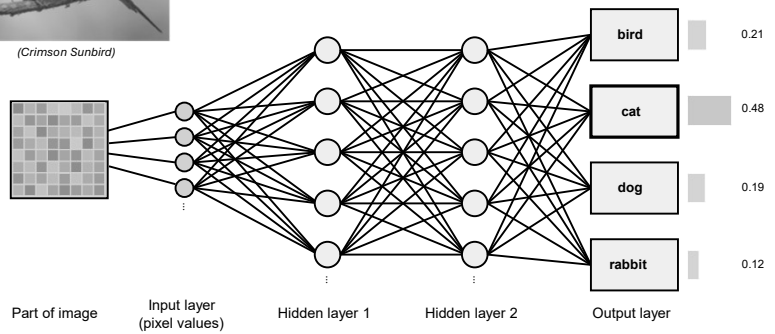


$(\text{weighted sum})\ z = b + x_1w_1 + x_2w_2 + x_3w_3 \quad \text{if } z > 0, \text{output} = z;\ \text{otherwise} = 0\ \ (\text{ReLU function})$

23

23

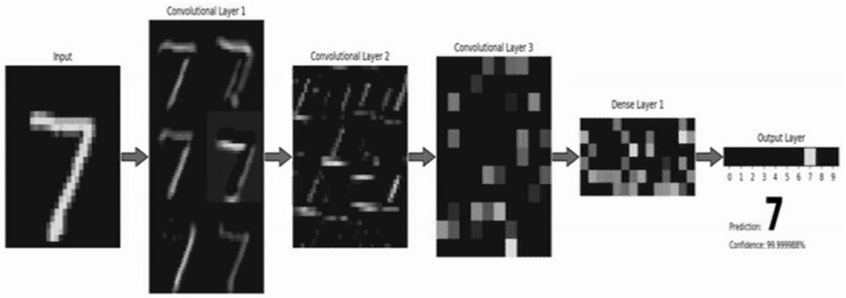

(Crimson Sunbird)



24

24

Patterns of activity in a digit-recognition network



25

25

Inscrutability and the Contrast with Classical AI

The information in a deep network is stored implicitly in the weights of its billions of connections. Its structure is not hand-crafted by any software engineer, nor controlled by any master program. It emerges from the multitude of patterns of activation passing through the layers from the input to the output and back (as learning occurs). The result is not humanly comprehensible even to an expert, and we are commonly left to "understand" and predict what the network does only by experience.

This is fundamentally different from the programs of Classical AI, whose data structures and algorithms are carefully crafted to structure and store the relevant information, and to process it accordingly. The computations of classical chess engine like Deep Blue (which beat Garry Kasparov in 1997) might be too *complex* for us to follow it in practice, but they are not fundamentally *inscrutable* in the way that a deep network is.

26

26

4. Are we Heading towards Conscious AI?

Turing considered his own original question "Can Machines Think?" to be "too meaningless to deserve discussion", but predicted that by 2000 "one will be able to speak of machines thinking without expecting to be contradicted" (1950, p. 442). This seems right, e.g.:

> "Why is the chess program taking so long to respond?"
>
> "It's *thinking* hard, because it's *realised* that if it *tries* to defend against my attack by bringing its knight over to protect the king, I'll be able to grab its pawn on the other side. It's displaying now that it *assesses* the position as better for me materially, but it's *predicting* that it won't be too badly off if it *decides* to let the pawn fall, so I think it must be *expecting* to get some activity to compensate."

27

27

Thinking and Understanding

Speaking after 2000, I agree that machines can think. The Tic-Tac-Toe program analyses any position in a way that is in essence structurally identical to how a human might do it. A human calculating a move in this way is said to be "thinking", and the computer running the program is without any doubt *calculating a move*. Why then deny that it is *thinking*?

> "Because it acts automatically, without *understanding* what it's doing."

But the notion of *understanding* here is at least as vague as *thinking*. The program's symbols link systematically to the board position and the rules of Tic-Tac-Toe, and its move is chosen logically so as to optimise its result in the game. A person whose mind was working in this way would be said to have a good understanding of the game (at least, much better than that of a hopeless novice) – why not say the same of the computer?

28

28

Reality versus Simulation

Here the sceptic might respond something like this:

> "The computer only *simulates* understanding – it's not the real thing. For example, it couldn't tell you what a game is, or why it's playing."

The latter is true enough, but nowadays we can easily imagine linking the program up with a chatbot or LLM to handle such interactions:

> "CERTAINLY I CAN TELL YOU WHAT A GAME IS … MY REASON FOR PLAYING TIC-TAC-TOE IS THAT I WAS PROGRAMMED TO DO SO. AND I REALLY ENJOY IT!"

There are three sentences here, the ***first*** of which is probably true (the program can indeed say what a game is); the ***second*** of which is dubious (can the program really be said to be undertaking actions for a "reason"?); while the ***third*** is downright false – the computer does not *enjoy* playing, any more that ELIZA is happy when "she" responds "I AM HAPPY".

29

29

When Simulation *is* Reality

Notice the crucial asymmetry here between *calculation* and *feeling*:

If the computer works out the best move in a game, in the same sort of way as an expert human, then it is *genuinely* calculating and identifying the best move. Likewise, if it solves a maths problem by a correct method and prints the right answer, then it has *genuinely* solved the problem.

But if the computer claims to *feel* something, that is quite different – because feeling involves a concrete reality beyond the symbolic output.

- Sufficiently good simulation of correct calculation – in terms of input and output behaviour – ***just is*** correct calculation.
- But no matter how convincing a computer's simulation of emotional "input" and "output" behaviour might seem to be, this **is not the same** as having genuine emotions (and perhaps not even good evidence).

30

30

So … Can Machines Genuinely Think?

The answer here will depend on whether we view "thinking" as more like *calculation*, or more like *feeling*. And notice that this is a problem that has only arisen since Turing's time, and precisely because of his own work.

Words like "thought" and "intelligence" acquired their meaning in a world that essentially contained two salient kinds of thing:

> *Agents* processing information about their perceived situation (using thoughts that represent the objects in their environment and other things), so as to adapt their behaviour to achieve their own purposes.
>
> *Inanimate objects* with no purposes of their own, no perception of their situation, and no processing of thoughts or other representations.

Turing's revolution brought something quite new into the world: an inanimate object that "intelligently" processes information!

31

31

Information Processing versus Phenomenology

I prefer to consider "Intelligence" as a measure of *sophisticated information processing* rather than *phenomenology* (i.e. what it's like from the inside). This fits with our thinking of intelligence as:

- Typically subject-relative;
- A matter of degree, not all-or-nothing;
- Measured by *performance*, *appropriateness*, and *flexibility*;
- Not significantly correlated with how strongly one feels;
- Often exhibited expertly with little conscious thought (e.g. driving).

When distinguished in this way, something can be *intelligent* without being *conscious* at all, and something can be *conscious* without having much *intelligence* (though *zero* seems evolutionarily unlikely).

32

32

Are We Heading Towards Conscious Machines?

Summing up, I'm happy to accept that we can potentially have *intelligent* machines (including modern LLMs), but – given the clear distinction that I am recommending – this does *virtually nothing whatever* to support the claim that they are potentially *conscious*.

In brief, the argument goes roughly like this:

1. Consciousness is a causal process, that has evolved in animals because it is highly conducive to our survival (especially in socially complex species) and reproduction.
2. There is no "causal space" for consciousness in the activity of digital computers – what they do is entirely determined by fundamental digital operations, with no hint of consciousness.

33

33

Consciousness and Causation

Common experience tell us that conscious experiences strongly influence our behaviour (and are impacted by what happens to us).

Moreover consciousness seems to be finely tuned to track things that have an impact on our evolutionary success, which would be a massive coincidence if it had no causal impact

- Conscious pleasures are strongly correlated with things that were evolutionarily beneficial in our ancestral environment (e.g. physical health, calories, social intimacy).
- Conscious pains are strongly correlated with things that have been evolutionarily harmful (e.g. injury, starvation, disease, ostracism).

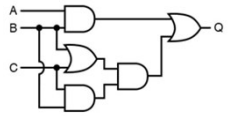
Evolution cannot act on traits that are causally inert.

34

34

What Ultimately Drives Computer Programs?

Computer programs are standardly written in a *high-level programming language* (e.g. Python), but when they run, this is translated – or *compiled* – into low-level *machine code*. The machine code instructions are themselves *decoded* by the computer's central processing unit, and ultimately generate signals that activate circuits of *logic gates* (as illustrated here).



The *physics* of the logic gates might be mysterious at the lowest level (e.g. quantum mechanics and materials). But the *abstract behaviour* of the logic gates can be modelled mathematically, and they are engineered with redundancy and error correction, so that the physics is not allowed to "interfere" with this abstract behaviour. Hence in a sense, *we have complete understanding of how the computational processes are operating*.

35

35

Thus in the computation of the computer's high-level output, there is *no causal room* for *mysterious causal factors* (such as consciousness). So to suppose that an LLM is conscious just because it emits language indicative of feelings is as gratuitous as to suppose that ELIZA is happy when "she" emits the text "I AM HAPPY". *We know* (at least in broad outline) what sorts of mechanisms have brought about the appearance of that text, and *we know* they have *nothing* to do with being conscious.

By contrast, when we are dealing with other humans (and similar animals), we have little clue about the fundamental mechanisms within their brains, any more than our own. But on the other hand, they are our kith and kin, evolved in the same broad family moulded by a common evolutionary history. So we have excellent reason to think that other people function biologically in broadly the same way as we do, *with consciousness playing a vital causal role*.

36

36