

# A Journey Through the History of Chess Programming

Or: “What Computers Can Do”

# Why is Chess so interesting?



# Maybe Chess can only be played by humans?

An observation – the best chess players report that they often rely purely on “intuition” when making decisions – some moves just “feel right”, with no further explanation forthcoming.

- Computers can only do symbol manipulation – arithmetic, logic, etc.
- Intuition is clearly not a form of symbol manipulation.

It seems as if it is *impossible* for computers to play chess!

Okay, maybe it's possible to know which move is best.

In 1928, John von Neumann proved that the Minimax algorithm was the optimal solution for games like Chess.

This still leaves us with the problem of *state-space search* – this is alternatively referred to as *the problem of branching factor*.

If a computer were to *really* play chess, it would have to somehow discard huge numbers of moves based on some more intelligent principles.

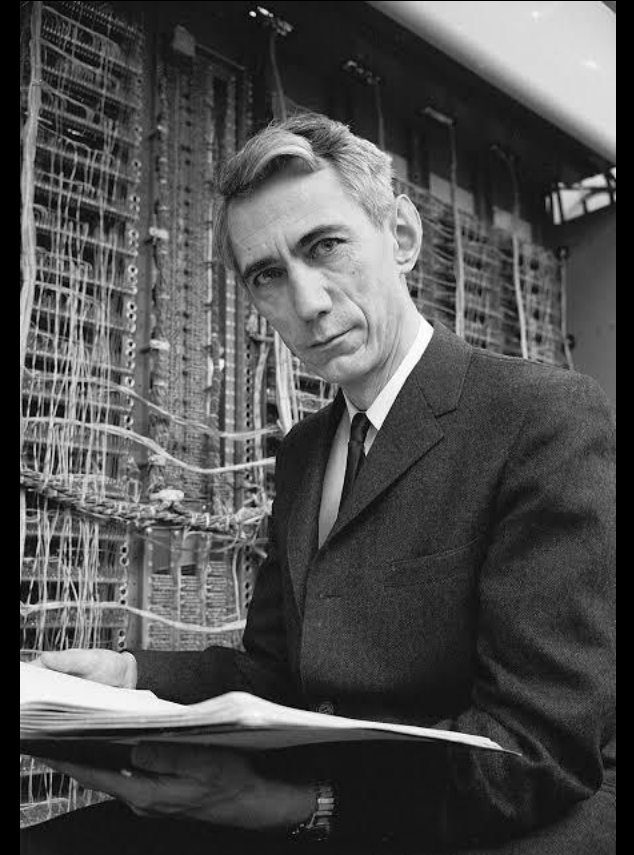


John von Neumann,  
wondering which scientific  
field he should invent next

# Which principles, then? And how intelligent must they be?

*“Although in chess there is no known simple and exact evaluating function  $f(P)$ , and probably never will be because of the arbitrary and complicated nature of the rules of the game, it is still possible to perform an approximate evaluation of a position.*

*Any good chess player must, in fact, be able to perform such a position evaluation. Evaluations are based on the general structure of the position, the number and kind of Black and White pieces, pawn formation, mobility, etc.”*



Claude Shannon, taking a short break from inventing all of information theory.

# Which principles, then? And how intelligent must they be?

Unfortunately, all obvious heuristics fail in many circumstances!

*“Most of us know how to ride a bicycle. Does that mean we can formulate specific rules to teach someone else how to do it? ... And do we really know, until the situation occurs, just what we would do in response to a certain wobbly feeling? No, we don’t.”*  
– Prof. Hubert Dreyfus

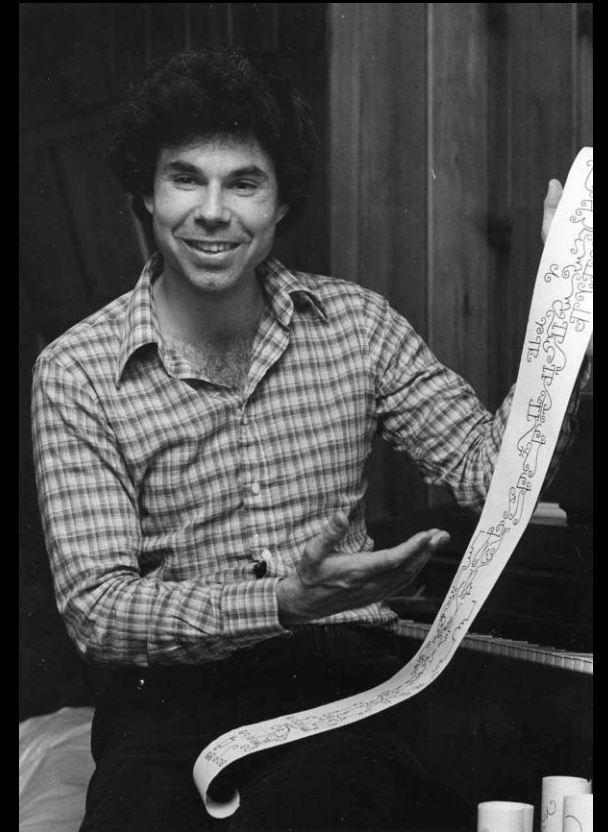


Hubert Dreyfus, *not* manipulating symbols in his head.

# What do we need to do, then?

In 1979, Douglas Hofstadter, philosopher, physicist and the author of *Gödel, Escher, Bach: An Eternal Golden Braid* thought that while super-human Chess-playing programs might very well be possible,

*"They will be programs of general intelligence, and they will be just as temperamental as people. "Do you want to play chess?" "No, I'm bored with chess. Let's talk about poetry." That may be the kind of dialogue you could have with a program that could beat everyone. That is because **real intelligence** inevitably depends on a total overview capacity."*



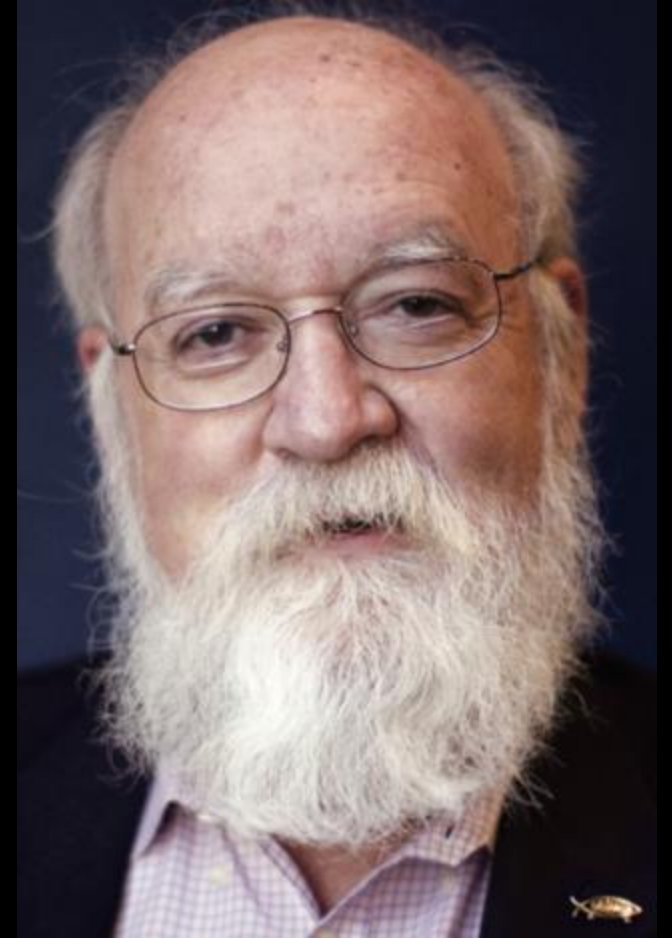
Doug Hofstadter,  
looping strangely.

# Real Intelligence? What's that?

A short detour: What is Real Magic?

*The analogy can be highly illuminating, as the following account by Lee Siegel on the reception of her work on magic can illustrate it: "I'm writing a book on magic", I explain, and I'm asked, "Real magic?" By real magic people mean miracles, thaumaturgical acts, and supernatural powers. "No", I answer: "Conjuring tricks, not real magic". Real magic, in other words, refers to the magic that is not real, while the magic that is real, that can actually be done, is not real magic.*

– Daniel Dennett, *Explaining The "Magic" of Consciousness*



Daniel Dennett, my favourite philosopher

# Real Intelligence? What's that?

So, when the machine finds Checkmate-in-Two in a middle-game position, it is merely “calculating” – the line of play was within reach of brute-force search.

Similarly, when the machine plays a perfect Rook and Pawn endgame, there are so few pieces that it can calculate all possible outcomes to the end of the game – not **Real Intelligence**, just brute force!

This is an easy thought to think when all the programs are simple endgame solvers and tactic finders, and none of them can yet play a proper game of chess – but this could not last forever.

# The First Steps

*1951* – Alan Turing produces the first true chess program, *Turochamp*, which he executed on paper and pencil, for the computers of the time are too weak to run it.

*1956* – John McCarthy invents the Alpha-Beta optimisation.

*1962* – The first program to play credibly, *Kotok-McCarthy*, is published at MIT.

*1967* – *Mac Hack VI* is released, with many enhancements over predecessors. It becomes the first program to defeat a person in tournament play.

# The Dreyfus Match

In 1965, Professor Hubert Dreyfus published *Alchemy and AI*, in which he talks of an “elaborate chess-playing program”, that “played poor but legal chess, and in its last official bout was beaten in 35 moves by a ten-year old novice.”

In 1967, several MIT students and professors (organized by Seymour Papert) challenged Dreyfus to play a game of chess against their program, *Mac Hack VI*.

Dreyfus accepted.

# The Dreyfus Match

Dreyfus was playing against *Mac Hack VI*.

“*Mac*” comes from MIT’s Project MAC (Mathematics And Computation), but was backronym’d to *Multiple Access Computer*, *Machine Aided Cognitions*, and *Man And Computer*.

The VI suffix is a reference to the PDP-6, which was the computer that MH6 ran on.



# The Ascent

After this, computer chess programs improved, but slowly.

1968 – Scottish chess champion David Levy makes a £500 bet with AI pioneers John McCarthy and Donald Michie that no computer program would win a chess match against him within 10 years.

1971 – Ken Thompson, creator of Unix, writes his first chess-playing program called "chess" for the earliest version of the OS.

1974 – David Levy et al organize the first WCCC, which is won by the Russian program *Kaissa*.

1977 – The first dedicated chess computer, *Chess Challenger*, is sold.

1978 – David Levy wins his bet, defeating *Chess 4.7* in a six-game match by a score of 4½–1½. The computer's victory in game four is the first defeat of a human master in a tournament.

# The Shape Of Things To Come



*Photo 5: The computer decides not to trade Rooks. Levy pulls his Bishop back to act as a shield against the final assault. He smiles as he sees that the steamroller pawns will not stop.*



# The Final Defeat of Humanity

## Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

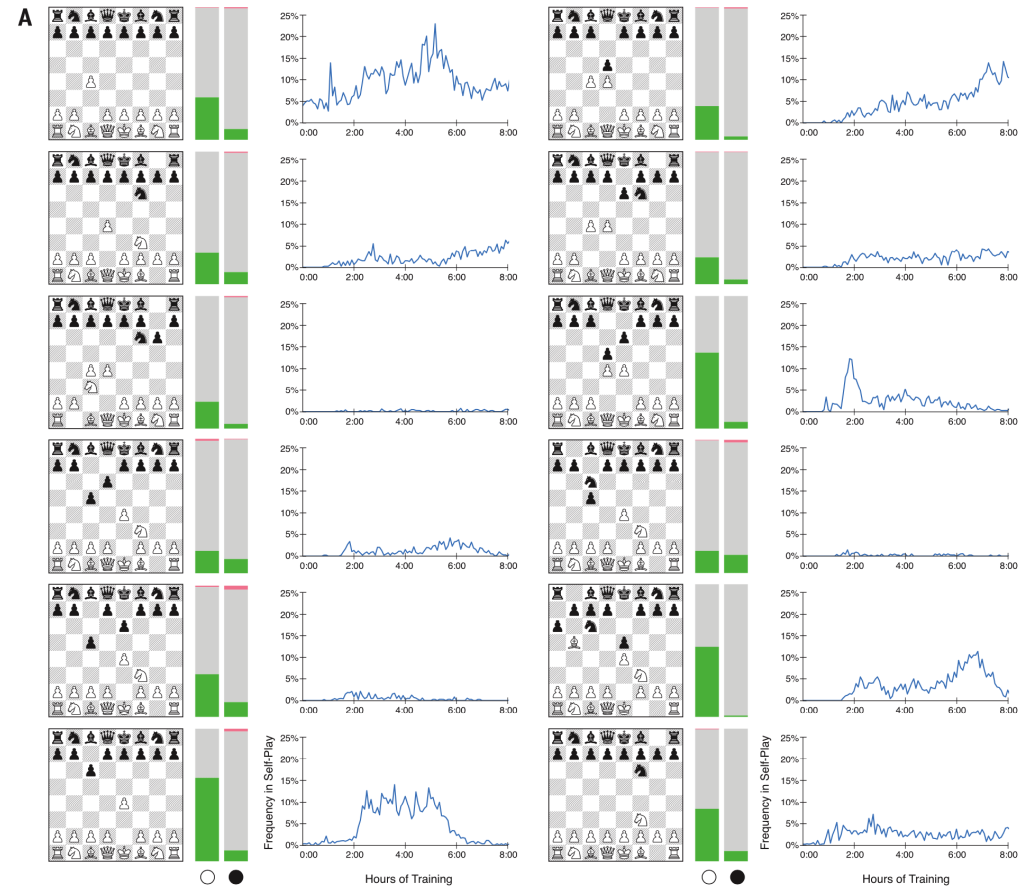
David Silver,<sup>1\*</sup> Thomas Hubert,<sup>1\*</sup> Julian Schrittwieser,<sup>1\*</sup>  
Ioannis Antonoglou,<sup>1</sup> Matthew Lai,<sup>1</sup> Arthur Guez,<sup>1</sup> Marc Lanctot,<sup>1</sup>  
Laurent Sifre,<sup>1</sup> Dharshan Kumaran,<sup>1</sup> Thore Graepel,<sup>1</sup>  
Timothy Lillicrap,<sup>1</sup> Karen Simonyan,<sup>1</sup> Demis Hassabis<sup>1</sup>

<sup>1</sup>DeepMind, 6 Pancras Square, London N1C 4AG.

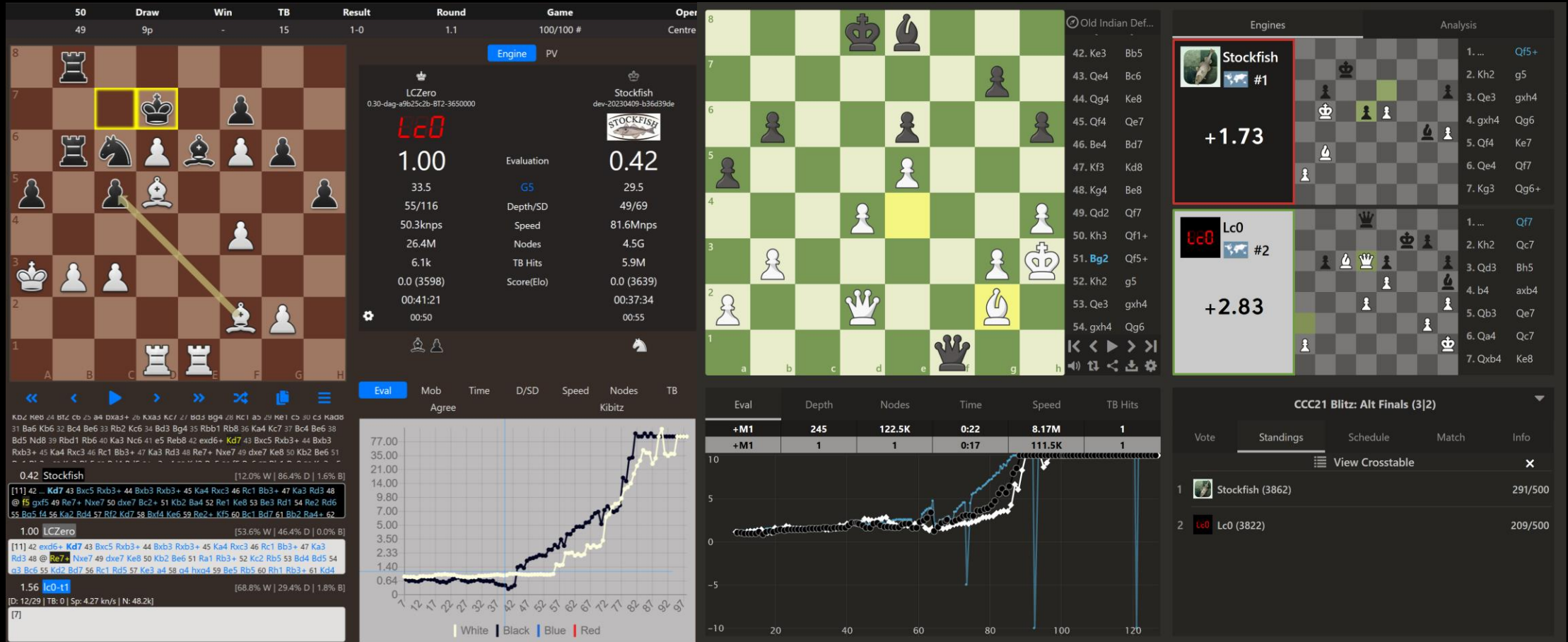
\*These authors contributed equally to this work.

### Abstract

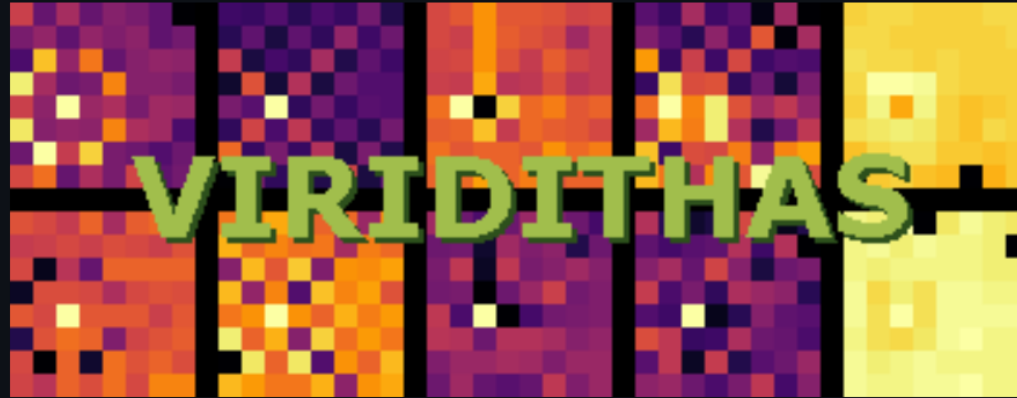
The game of chess is the most widely-studied domain in the history of artificial intelligence. The strongest programs are based on a combination of sophisticated search techniques, domain-specific adaptations, and handcrafted evaluation functions that have been refined by human experts over several decades. In contrast, the *AlphaGo Zero* program recently achieved superhuman performance in the game of Go, by *tabula rasa* reinforcement learning from games of self-play. In this paper, we generalise this approach into a single *AlphaZero* algorithm that can achieve, *tabula rasa*, superhuman performance in many challenging domains. Starting from random play, and given no domain knowledge except the game rules, *AlphaZero* achieved within 24 hours a superhuman level of play in the games of chess and shogi (Japanese chess) as well as Go, and convincingly defeated a world-champion program in each case.



# Modern Computer Chess



# Viridithas, a UCI chess engine written in Rust

 BUILD

PASSING

LICENSE

MIT

OFFICIAL RELEASE

V10.0.0

COMMITTS SINCE V10.0.0

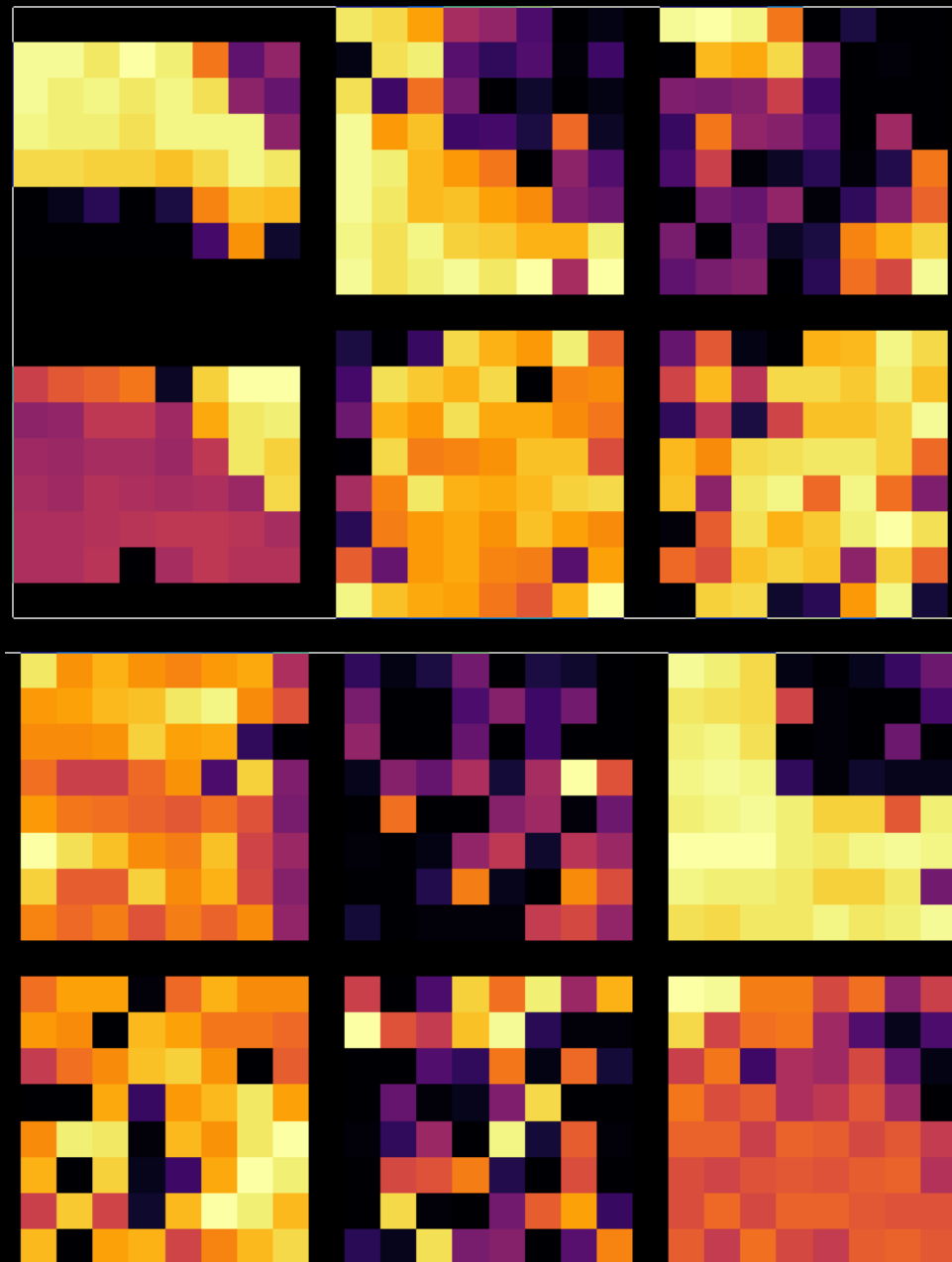
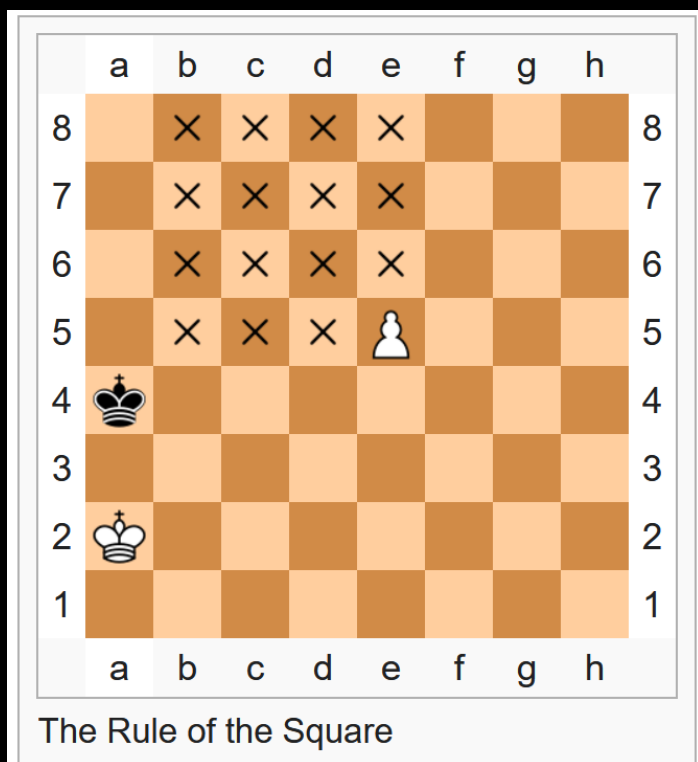
51

Viridithas is a free and open source chess engine and the successor to the original [Viridithas](#).

Viridithas is not a complete chess program and requires a UCI-compatible graphical user interface in order to be used comfortably. Read the documentation for your GUI of choice for information about how to use Viridithas with it.

For an overview of the features of Viridithas, see the [viri-wiki](#).

# Neural Network Interpretability



# Takeaways

- When a computer does something *really impressive*, it's not magic!
- When computers get good at something, they rapidly overtake human performance.
- Neural networks are not necessarily black boxes, interpretability work can be really effective!

And most importantly -

- When a scientist or an intellectual says that something *cannot be done*, this is *no reason at all* to think that it won't be!

# Why general artificial intelligence will not be realized

[Ragnar Fjelland](#) 

[Humanities and Social Sciences Communications](#) **7**, Article number: 10 (2020) | [Cite this article](#)

**174k** Accesses | **93** Citations | **766** Altmetric | [Metrics](#)

## Abstract

The modern project of creating human-like artificial intelligence (AI) started after World War II, when it was discovered that electronic computers are not just number-crunching machines, but can also manipulate symbols. It is possible to pursue this goal without assuming that machine intelligence is identical to human intelligence. This is known as weak AI. However, many AI researcher have pursued the aim of developing artificial intelligence that is in principle identical to human intelligence, called strong AI. Weak AI is less ambitious than strong AI, and therefore less controversial. However, there are important controversies related to weak AI as well. This paper focuses on the distinction between artificial general intelligence (AGI) and artificial narrow intelligence (ANI). Although AGI may be classified as weak AI, it is close to strong AI because one chief characteristics of human intelligence is its generality. Although AGI is less ambitious than strong AI, there were critics almost from the very beginning. One of the leading critics was the philosopher Hubert Dreyfus, who argued that computers, who have no body, no childhood and no cultural practice, could not acquire intelligence at all. One of Dreyfus' main arguments was that human knowledge is partly tacit, and therefore cannot be articulated and incorporated in a computer program. However, today one might argue that new approaches to artificial intelligence research have made his arguments obsolete. Deep learning and Big Data are among the latest approaches, and

Download PDF



## Associated Content

Collection

**Ethics of quantification: big data and governing through numbers**

Sections

References

[Abstract](#)

[Introduction](#)

[Tacit knowledge](#)

[Three "milestones" in AI research](#)

[The latest hype: Big Data](#)

[Correlations and causes](#)

[The mini Turing test](#)

[The real Turing test](#)

[Conclusion: computers are not in the world](#)

[Notes](#)

[References](#)

[Acknowledgements](#)

[Author information](#)

[Ethics declarations](#)

# Resources / Links

- [The Chess Programming Wiki](#)
- The [Top Chess Engine Competition](#)
- Chess.com's [Computer Chess Championship](#)
- My program's [GitHub Repository](#)
- [Mechanistic Interpretability for Neural Networks](#)