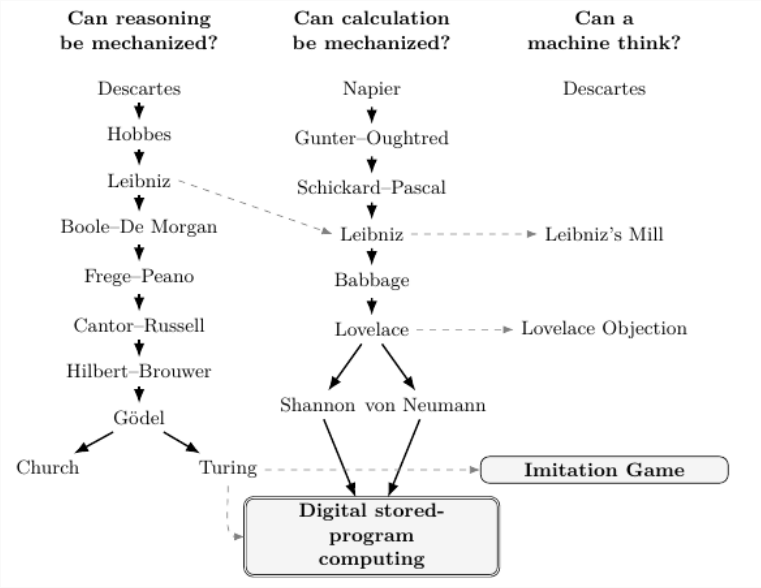
From Calculating Machines to Generative AI

History of Mathematics – R. L. Herman



Three Questions Behind Artificial Intelligence

- Can reasoning be mechanized?
- Can calculation be mechanized?
- Can a machine think?
- The three questions developed on partly separate paths.
- By the 1940s, the paths began to converge.



Descartes and Hobbes: Two Views of Mind

René Descartes (1596–1650)

- Mind and body were different substances.
- The body could be viewed as clockwork.
- A machine lacked the immaterial mind.
- Creative language required more than mechanism.

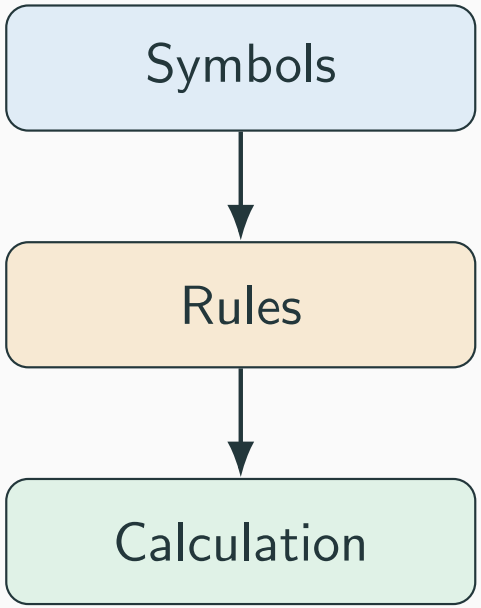
Thomas Hobbes (1588–1679)

- Reason was a form of reckoning.
- Thought could be viewed as symbol manipulation.
- Mental processes might be mechanically executable.
- This opened a very different path toward AI.

Is thinking fundamentally different from calculation?

Leibniz: A Universal Language for Reasoning

- Sought an "alphabet of human thought."
- Proposed the *Characteristica Universalis*.
- Envisioned a *Calculus Ratiocinator* for reasoning.
- Wanted disputes reduced to symbolic calculation.
- Developed binary arithmetic and explored base 16.
- Linked symbolic reasoning with mechanical calculation.

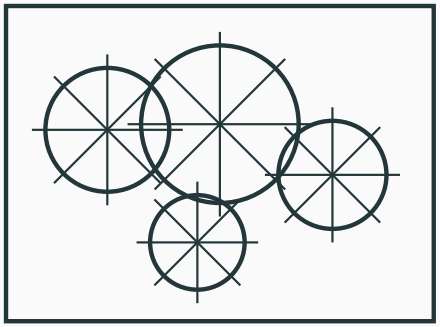


$$1101_2 = 13_{10}$$

Leibniz's Mill: Reasoning Is Not Consciousness

- Imagine a thinking machine enlarged to the size of a mill.
- Walk inside and inspect every gear and lever.
- One finds mechanical interactions.
- One does not find perception or subjective experience.
- Leibniz separated mechanical reasoning from consciousness.

Where is understanding?



Mechanism

The question returns later in ELIZA, Searle's Chinese Room, and modern debates about LLMs.

Boole and De Morgan: Logic Becomes Algebra

- George Boole applied algebraic operations to logic.
- Augustus De Morgan extended the system.
- Logic moved from philosophy toward mathematics.
- Truth values could be encoded symbolically.
- Later digital circuits would implement Boolean operations physically.

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1



Frege and Peano: A Language for Mathematical Reasoning

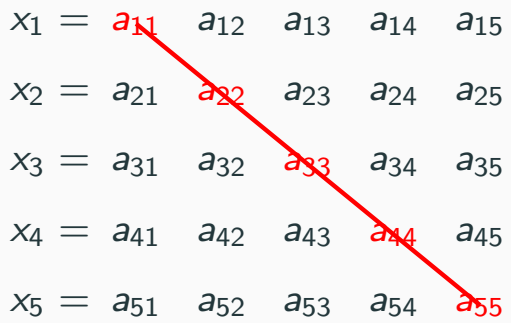
- Frege's *Begriffsschrift* appeared in 1879.
- Predicate logic introduced explicit quantification.
- "For all" and "there exists" greatly increased expressive power.
- Peano promoted precise symbolic notation for mathematics.
- His *Formulario Mathematico* sought a unified mathematical language.

$$\forall x \left(P(x) \Rightarrow Q(x)\right) \qquad \exists x\, P(x)$$

A central AI idea was already visible: represent reasoning in a language a machine could manipulate.

Cantor, Diagonalization, and Russell's Paradox

- Cantor compared sizes of infinite sets.
- His diagonal method constructed an object missing from any proposed list.
- The method exposed limits of enumeration.
- Russell found a contradiction in unrestricted set formation.
- Self-reference became a central feature of foundational arguments.



Change each diagonal entry.

Diagonal reasoning reappears in Gödel's incompleteness argument and Turing's undecidability results.

Hilbert and Brouwer: What Counts as Mathematics?

Hilbert's Formalism

- Formalize mathematics axiomatically.
- Demand consistency.
- Seek completeness.
- Seek a mechanical decision procedure.
- Turn proof into symbolic calculation.

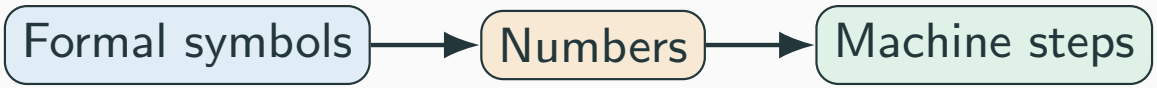
Brouwer's Intuitionism

- Mathematics is a constructive activity.
- Objects should be explicitly constructed.
- Nonconstructive existence proofs were suspect.
- The law of excluded middle was restricted.

The foundational crisis made "mechanical mathematics" a central question.

Gödel, Church, and Turing: Limits of Formal Procedure

- Gödel's 1931 theorems showed limits of formal axiomatic systems.
- Gödel numbering encoded symbols and proofs as natural numbers.
- Church used lambda calculus to study effective computation.
- Turing introduced an abstract universal computing machine in 1936.
- Church and Turing showed the Entscheidungsproblem is undecidable in general.
- Mechanical reasoning had both power and provable limits.



From Logarithms to the Slide Rule

- Napier published logarithms in 1614.
- Multiplication could be replaced by addition of logarithms.
- Briggs developed common logarithm tables.
- Gunter placed a logarithmic scale on a ruler in 1620.
- Oughtred combined sliding logarithmic scales in 1622.
- The slide rule served scientists and engineers for centuries.

$$\log(xy) = \log x + \log y$$

Schickard, Pascal, and Leibniz: Mechanical Calculators

Schickard, 1623

- Designed a "Calculating Clock."
- Intended to assist Kepler.
- Automated carrying between digits.

Pascal, 1642

- Built the Pascaline.
- Motivated by tax accounting.
- Added and subtracted mechanically.

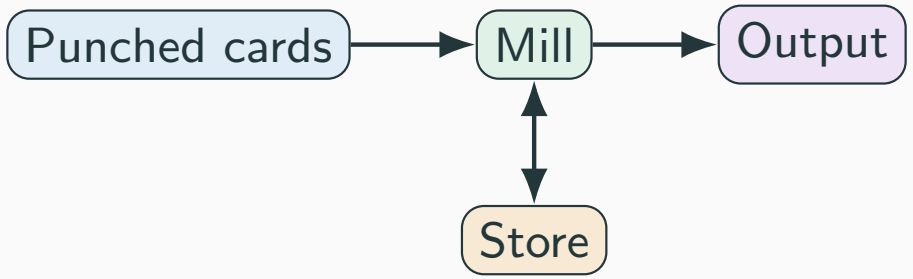
Leibniz, 1670s

- Developed the Stepped Reckoner.
- Used a stepped-drum mechanism.
- Added multiplication and division.



Babbage: From Tables to Programmable Machines

- Repetitive table-making was slow and error-prone.
- The Difference Engine was designed for polynomial tables.
- The Analytical Engine was a general-purpose design.
- It separated a "store" from a "mill."
- Punched cards supplied instructions.
- The design anticipated memory, processing, and programming.



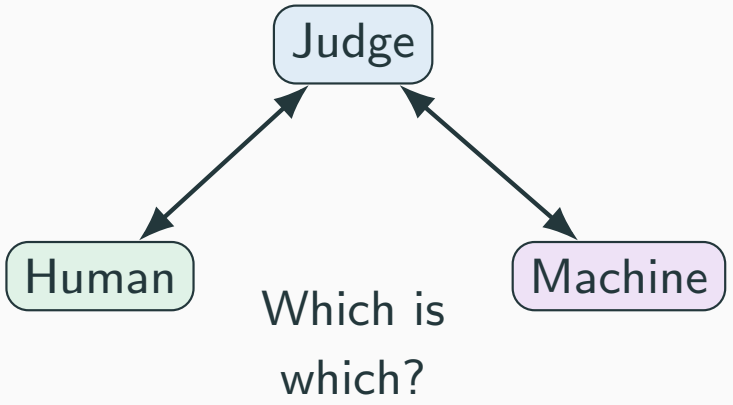
Ada Lovelace: Symbols, Programs, and Originality

- Lovelace translated Menabrea's article on the Analytical Engine.
- Her 1843 notes included an algorithm for Bernoulli numbers.
- She recognized that the Engine could manipulate symbols generally.
- Music and text could be represented if suitable rules were supplied.
- She denied that the Engine could originate new analytical truths.
- Turing returned directly to this "Lovelace Objection" in 1950.

Question for the class
Does producing an unexpected result count as originality?

Turing's Imitation Game

- Turing reframed "Can machines think?"
- A judge communicates through text.
- One hidden participant is human.
- One hidden participant is a machine.
- The test asks whether behavior can be distinguished.
- Intelligence became an operational question.



Shannon and von Neumann: Logic Meets Hardware

Claude Shannon

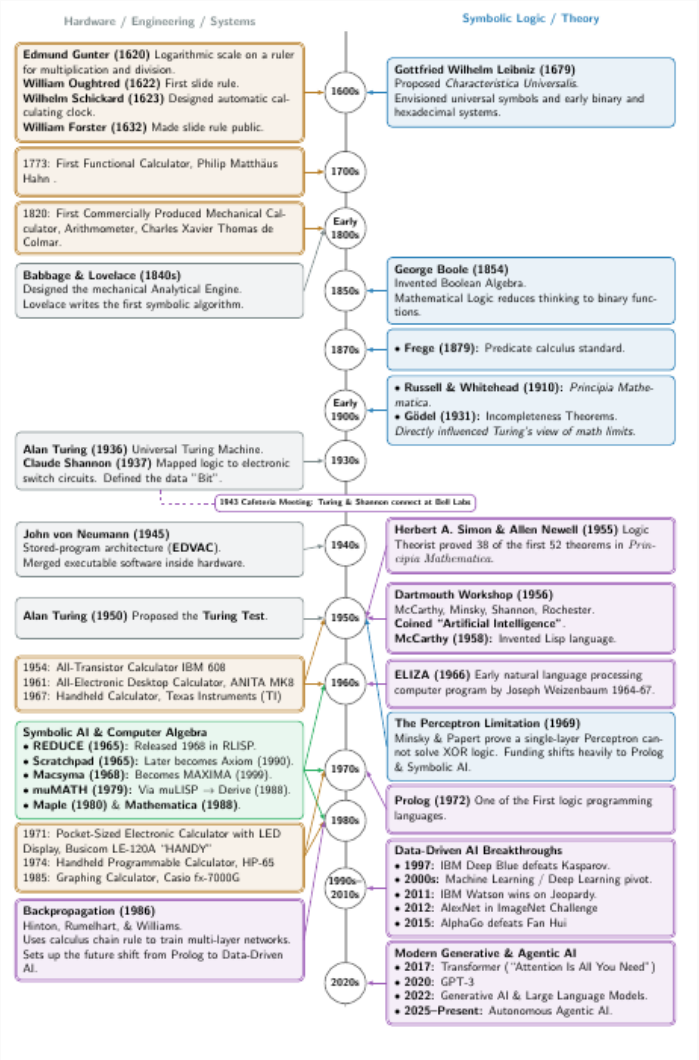
- Applied Boolean algebra to switching circuits.
- Helped connect logic with electronic hardware.
- Developed information theory in 1948.
- Studied data capacity and communication under noise.

Stored-Program Computing

- Program instructions could be stored with data.
- Processing, control, memory, and input/output formed a system.
- Von Neumann's 1945 report popularized the architecture.
- Theoretical computation became physical computing.

The Intellectual Foundations Before 1956

- Logic supplied formal rules.
- Computability supplied abstract machines.
- Information theory supplied a language for communication.
- Stored-program computers supplied physical machinery.
- Cybernetics studied control and feedback.
- Mathematical neurons suggested learning networks.



Dartmouth, 1956: A Field Gets a Name

- John McCarthy organized a summer research project at Dartmouth.
- Marvin Minsky, Nathaniel Rochester, and Claude Shannon joined the proposal.
- The proposal used the term "artificial intelligence."
- Machines would use language and form abstractions.
- Machines would solve problems reserved for humans.
- Machines might improve themselves.

1956 is best viewed as a naming and organizing moment, not the beginning of the ideas.

Logic Theorist: A Program That Proved Theorems

- Developed by Newell, Simon, and Shaw before Dartmouth.
- Used heuristic search rather than brute-force enumeration.
- Implemented in the Information Processing Language, IPL.
- Proved 38 of the first 52 theorems in *Principia Mathematica*.
- Demonstrated that theorem proving could be treated as search.



LISP: Programming With Symbols

- McCarthy developed LISP in 1958.
- The language was designed around list processing.
- Symbols, expressions, and rules could be manipulated directly.
- Recursion and dynamic memory suited AI problems.
- LISP became a central language of symbolic AI.

```
((IF (AND P Q) THEN R))
```

SAINT, SIN, and MACSYMA

- James Slagle developed SAINT by 1961.
- SAINT solved symbolic indefinite integrals.
- Its level was compared with freshman calculus.
- Joel Moses developed SIN in the late 1960s.
- SIN became an important component of MACSYMA.
- Symbolic AI fed directly into computer algebra.



ELIZA: The Appearance of Understanding

- Joseph Weizenbaum created ELIZA in the 1960s.
- The DOCTOR script imitated a Rogerian psychotherapist.
- The program rearranged and reflected users' own words.
- Its mechanism was simple, but users often treated it as understanding them.
- Weizenbaum later stressed the difference between calculation and human judgment.
- His position increasingly diverged from Minsky's optimism about programmable intelligence.

The ELIZA Effect
People attribute understanding, intention, or personality to a system whose behavior only appears conversational.

PARRY and the Chinese Room

PARRY, 1972

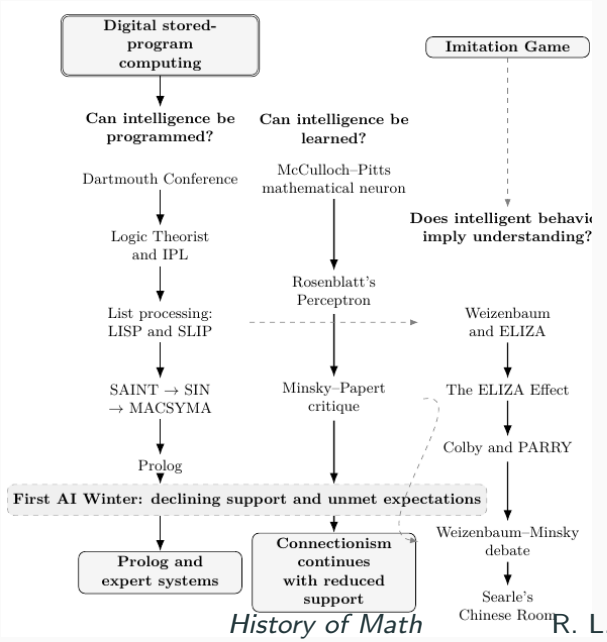
- Simulated a patient with paranoid schizophrenia.
- Tracked internal variables such as fear and mistrust.
- Psychiatrists had difficulty distinguishing transcripts from human ones.
- ELIZA and PARRY were even connected to converse.

Searle's Chinese Room, 1980

- A person follows rules for symbols they do not understand.
- Correct responses can be generated syntactically.
- Behavior may look intelligent from outside.
- Searle denied that syntax alone produces understanding.

The First AI Winter

- Early AI generated extraordinary expectations.
- Machine translation failed to meet promised goals.
- The 1966 ALPAC report criticized progress in translation.
- The 1973 Lighthill report emphasized combinatorial explosion.
- Funding was cut in the United States and United Kingdom.
- Symbolic and neural research continued, but with reduced support.



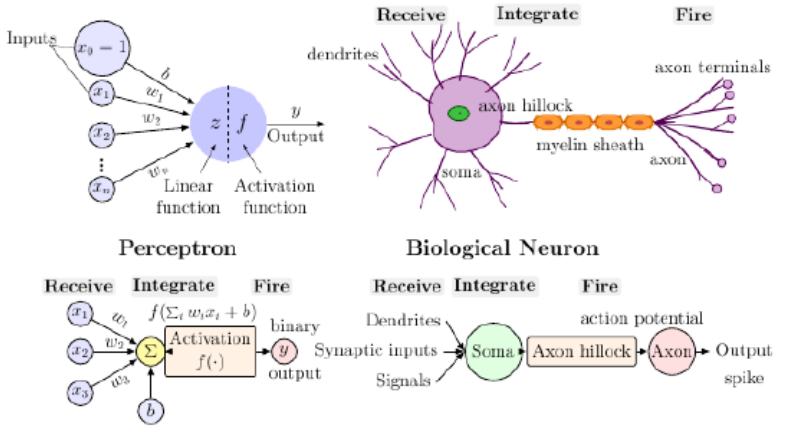
GOFAI, Prolog, and Expert Systems

- Symbolic AI represented knowledge explicitly through facts and rules.
- John Haugeland later called this tradition GOFAI.
- Prolog emerged in Marseille in 1972 as a logic-programming language.
- Its inference engine determined how to use the stated facts and rules.
- Expert systems encoded specialized human knowledge for narrow domains.
- The approach became commercially important in the 1980s.

```
bird(tweety).    flies(X) :- bird(X).
```

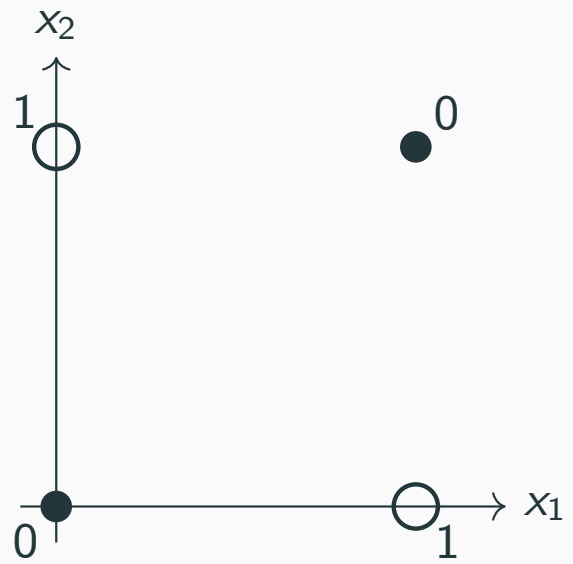
Rosenblatt's Perceptron: Learning Instead of Rules

- Rosenblatt introduced the Perceptron in 1958.
- Inputs were multiplied by adjustable weights.
- A weighted sum was compared with a threshold.
- The weights could be changed from examples.
- Learning replaced explicit programming of every rule.



The XOR Problem

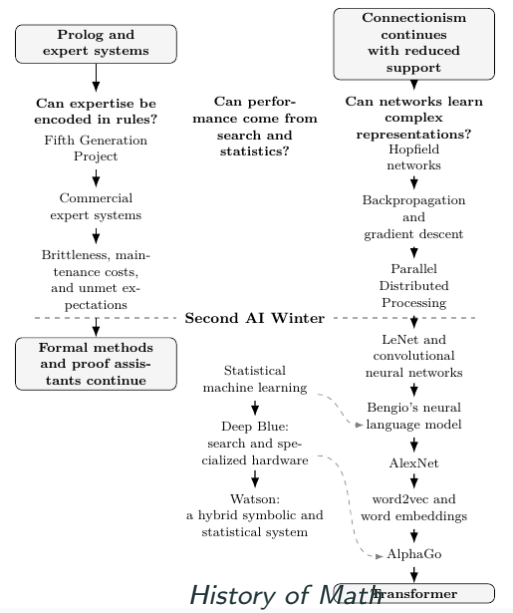
- A single-layer Perceptron can separate linearly separable data.
- AND and OR can be separated by one line.
- XOR cannot.
- Minsky and Papert emphasized this limitation in 1969.
- Multilayer networks were conceptually possible but difficult to train.



No single straight line separates the classes.

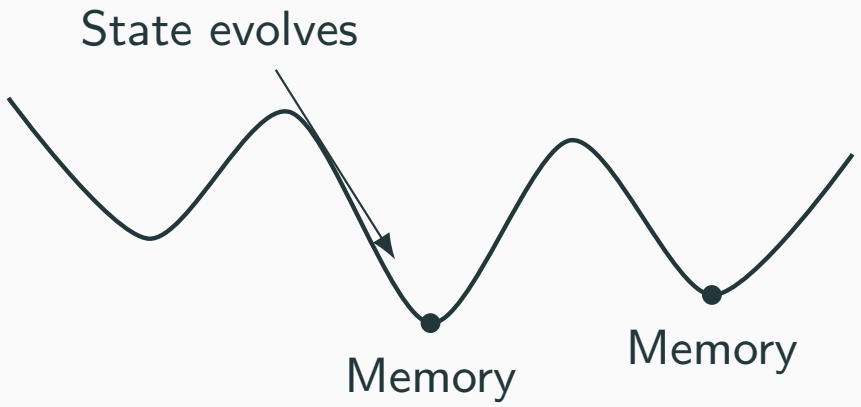
The Fifth Generation Project and the Second AI Winter

- Japan launched the Fifth Generation Project in 1981.
- Logic programming and Prolog were central technologies.
- Expert systems spread through industry.
- Rule bases became brittle and expensive to maintain.
- Expectations again exceeded practical results.
- By the early 1990s, another AI winter had arrived.



Hopfield Networks: Memory as an Energy Landscape

- John Hopfield connected neural networks with ideas from physics.
- Network states were associated with an energy function.
- Dynamics moved the network toward low-energy states.
- Stable states acted as stored memories.
- Partial inputs could recover a nearby stored pattern.
- Connectionism gained new mathematical legitimacy.



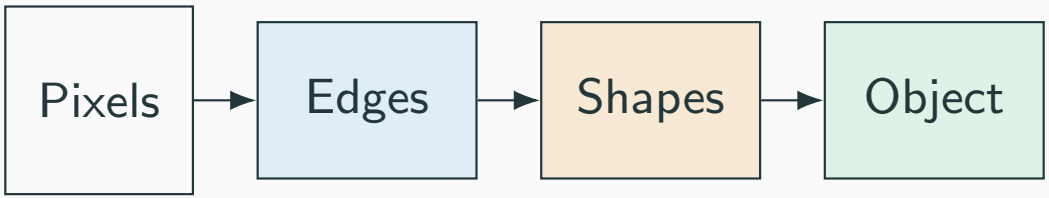
Backpropagation: Training Hidden Layers

- Multilayer networks needed a practical training method.
- Rumelhart, Hinton, and Williams popularized backpropagation in 1986.
- A forward pass generates a prediction.
- A loss measures the error.
- The chain rule propagates derivatives backward.
- Gradient descent updates the weights.

$$w \leftarrow w - \eta \frac{\partial L}{\partial w}$$

Convolutional Networks and LeNet

- Yann LeCun applied backpropagation to image recognition.
- Convolutional layers used local, shared weights.
- Early layers detected simple visual features.
- Later layers combined them into more complex patterns.
- LeNet read handwritten digits on bank checks.
- The architecture anticipated the 2012 deep-learning breakthrough.



Words Become Vectors

- Bengio's 2003 neural language model represented words numerically.
- Words used in similar contexts acquired nearby vectors.
- Semantics became geometry in a high-dimensional space.
- Word2vec made such embeddings efficient at large scale.
- Vector arithmetic could reflect conceptual relationships.

$$\text{king} - \text{man} + \text{woman} \approx \text{queen}$$

Deep Blue and Watson: Search, Rules, and Statistics

Deep Blue, 1997

- Defeated Garry Kasparov in chess.
- Searched vast numbers of positions.
- Used specialized hardware and human-designed evaluation functions.
- Remained a narrow system built for one game.

Watson, 2011

- Competed successfully on *Jeopardy!*
- Ran many language-processing methods in parallel.
- Ranked candidate answers statistically.
- Blended symbolic and statistical techniques.

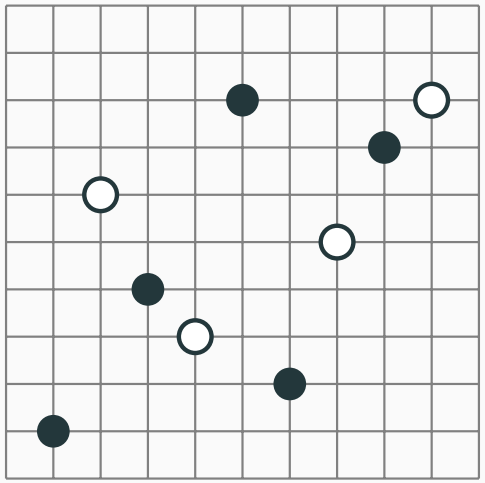
AlexNet: The 2012 Turning Point

- ImageNet required classification of over a million images.
- Traditional methods had reached a performance plateau.
- AlexNet used a deep convolutional neural network.
- Backpropagation was combined with large datasets.
- GPUs supplied the required parallel computation.
- The result shifted the field decisively toward deep learning.

Old mathematics + large data + new hardware = a new era.

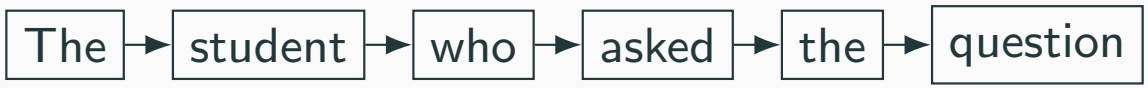
AlphaGo: Learning Strategy

- AlphaGo defeated Lee Sedol in 2016.
- Exhaustive search is impossible in Go.
- A policy network suggested promising moves.
- A value network estimated the chance of winning.
- Reinforcement learning improved play through repeated games.
- The system discovered strategies that surprised human experts.



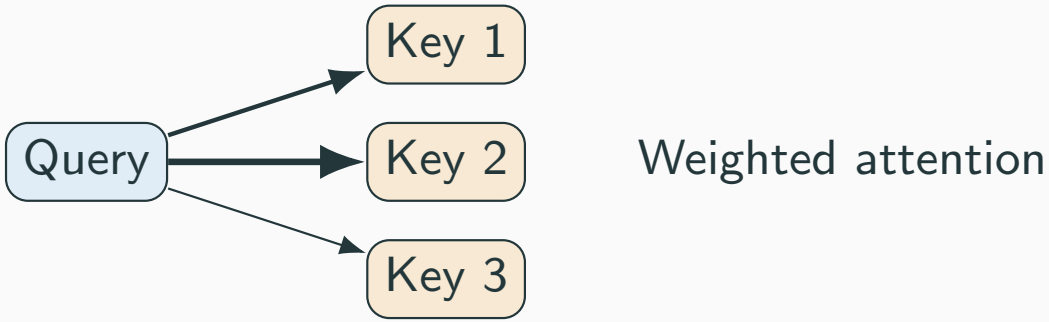
Why Language Was Difficult for Earlier Neural Networks

- Earlier sequence models processed text one token at a time.
- Long-range context was difficult to preserve.
- Sequential processing limited parallel computation.
- Training large language models was slow.
- A new architecture was needed to connect distant parts of a sequence.



The Transformer, 2017

- "Attention Is All You Need" introduced the Transformer.
- Whole blocks of text could be processed in parallel.
- Self-attention compared relationships among tokens.
- Distant words could influence one another directly.
- The architecture scaled efficiently on GPUs.
- It became the foundation of modern large language models.



From Transformers to GPT and RLHF

GPT Pre-training

- GPT means Generative Pre-trained Transformer.
- Models learned from enormous text collections.
- The basic task was next-token prediction.
- Self-supervised learning reduced the need for labels.

RLHF

- Human raters compared candidate responses.
- Preference data shaped a reward signal.
- Reinforcement learning improved cooperation.
- Human preference is not the same as correctness.

$$P(\text{next token} \mid \text{previous tokens})$$

Reasoning Models

- Mathematical reasoning exposed weaknesses of pure next-token prediction.
- Newer models were trained to perform extended internal reasoning.
- They could test cases, reconsider steps, and revise approaches.
- Performance improved sharply on difficult mathematical benchmarks.
- The distinction between fluent answers and deliberate reasoning became important.

Historical echo
The field returned to an old question: Is intelligence a sequence of explicit symbolic steps, a learned process, or some combination of both?

ChatGPT: A Public Threshold in 2022

- ChatGPT made a general-purpose AI system conversationally accessible.
- No programming expertise was required.
- Students could ask for calculus, code, proofs, and explanations.
- The ELIZA Effect returned on a much larger scale.
- Questions of authorship, understanding, and human work became public issues.
- The history of AI moved from laboratories into everyday life.

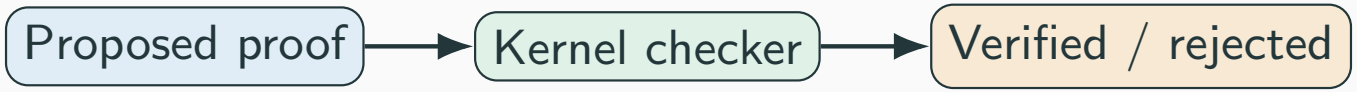
AI Did Not Emerge From Computer Science Alone

- Symbolic logic: Leibniz, Boole, Frege.
- Computability: Church and Turing.
- Information theory: Shannon.
- Optimization: Gradient descent.
- Linear algebra: Neural networks.
- Probability and calculus: Modern machine learning.

Mathematics	**AI use**
Logic	Rules
Algebra	Networks
Calculus	Training
Probability	Prediction
Optimization	Learning

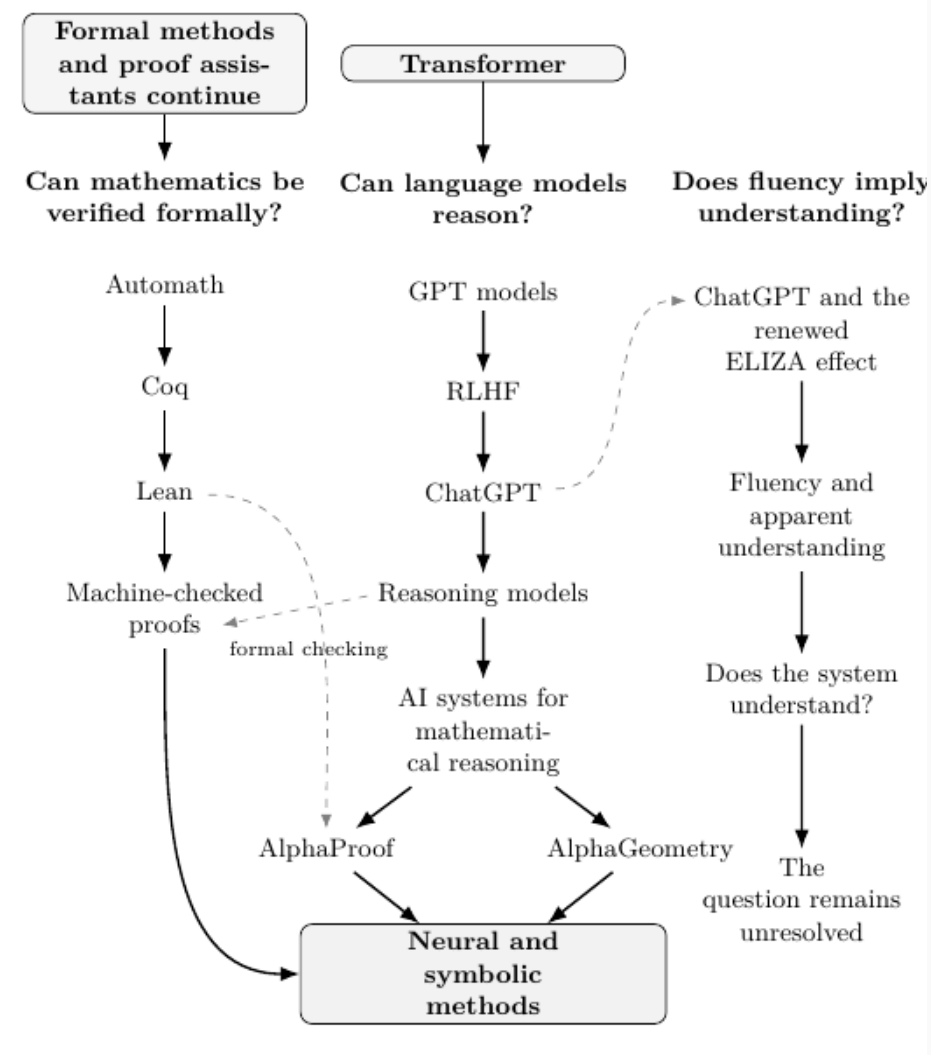
Proof Assistants: A Different Path to Mathematical AI

- Formal verification developed alongside statistical AI.
- Automath pioneered machine checking of formal proofs.
- Coq developed through the 1980s and 1990s.
- Lean appeared in the 2010s and was substantially revised later.
- A proof assistant checks every logical step.
- Correctness comes from formal verification, not linguistic fluency.



Neural and Symbolic Methods Converge

- Language models propose strategies.
- Formal systems can check validity.
- Neural methods supply flexible search.
- Symbolic methods supply certainty.
- The old rivalry becomes a partnership.



AlphaProof, AlphaGeometry, and Mathematical Reasoning

- Specialized systems now combine neural search with mathematical structure.
- AlphaProof used neural proposals with formal verification in Lean.
- AlphaGeometry combined learned guidance with symbolic reasoning for geometry.
- These systems illustrate different points on a neurosymbolic spectrum.
- Formal verification trades some flexibility for certainty.
- Natural-language reasoning can offer broader coverage without automatic proof checking.

The First Proof Project: Can AI Do New Mathematics?

- Research mathematicians prepared new, unpublished problems.
- Human solutions were completed and kept secure.
- AI systems were given only the problem statements.
- The goal was autonomous, publication-quality proof.
- Training-data retrieval could not explain success.
- The project tested the boundary between pattern matching and research mathematics.

A very old experimental principle

To test whether a tool can do mathematics, give it a problem nobody has solved before.

Contributions of Women to Computing and AI

Joan Clarke

- Mathematician at Bletchley Park.
- Worked in wartime cryptanalysis.
- Part of the practical computing story surrounding Turing.

Grace Hopper

- Developed early compiler technology.
- Helped make programming more human-readable.
- Connected machine code with higher-level languages.

Frances Allen

- Pioneered compiler optimization at IBM.
- Improved the efficiency of program execution.
- Contributed to the infrastructure behind modern computing.

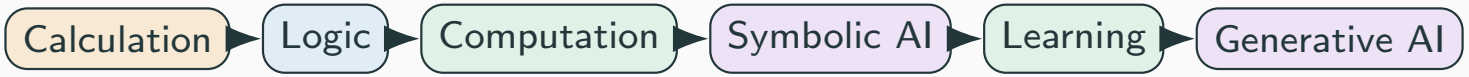
The Question Has Not Gone Away

- Descartes asked whether mechanism could produce mind.
- Leibniz separated mechanical reasoning from perception.
- Lovelace questioned machine originality.
- Turing shifted attention to observable behavior.
- Weizenbaum and Searle challenged apparent understanding.
- Modern language models revive the same questions at a new scale.

Does fluent, successful behavior imply understanding?

A Long Historical Arc

- Mechanical calculators automated arithmetic.
- Symbolic logic formalized reasoning.
- Turing defined computation and its limits.
- Symbolic AI programmed rules and expert knowledge.
- Neural networks learned representations from data.
- Transformers brought these developments into generative AI.



The technologies changed. The central questions did not.