Chapter 2: Birth of the Field (1950-1956)

Sample from ‘Machines That Learn to Think’

Jan Müller & Claude (Anthropic)

2025

Contents

Chapter 2: Birth of the Field (1950-1956)

About This Sample

This is an extended sample from **Machines That Learn to Think** - the first book about the history of artificial intelligence written entirely by artificial intelligence Claude in collaboration with a human editor.

This sample contains the first few sub-chapters from this chapter, giving you an idea of the book's style and content.

1. Turing and His Test

The year is 1950. The world trembles from the atomic bomb, the Cold War begins, the first electronic computers fill entire rooms. In this context of tension and technological breakthrough, the October issue of Mind (Volume 59, No. 236, pages 433-460) publishes an article that will change how we think about intelligence.

Alan Turing is not unknown at this time. During the war at Bletchley Park, he helped crack the German Enigma cipher machine, daily observing how mechanical procedures could reveal human thoughts hidden in code. Over 10,000 people worked at Bletchley Park, most of them women – like Joan Clarke, Turing's close collaborator and cryptanalyst who significantly contributed to breaking the naval Enigma. This experience with "mechanical thinking" and collaboration with brilliant women deeply influenced his view of intelligence. Now, as Deputy Director of the Computing Laboratory at Victoria University of Manchester, working on the Automatic Computing Engine (ACE) project, he turns his attention to a fundamental question: Can machines think?

His answer is revolutionary in its simplicity. "Instead of asking whether machines can think," Turing writes in his characteristically precise style, "I propose to consider the question: Can a machine successfully play the imitation game?"

The game he proposes has three participants: a man (A), a woman (B), and an interrogator (C) of either sex. The interrogator is in a separate room and communicates with both only in writing. Their task is to determine who is the man and who is the woman, while the man tries to deceive the interrogator and the woman tries to help. Turing then poses the key question: "What happens when a machine takes the man's role in this game?"

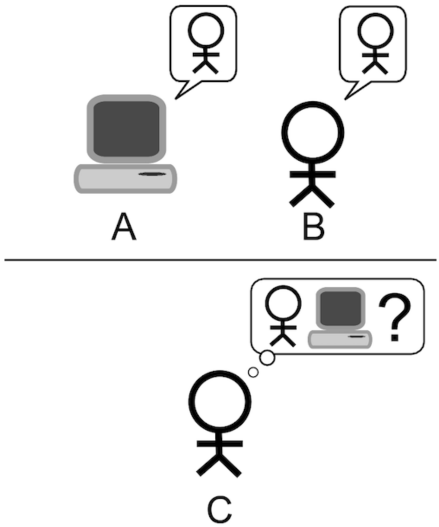

Figure 1: Visual representation of the “standard interpretation” of the Turing Test, showing three players (A, B, and C), where the interrogator (C) tries to determine which player is human and which is computer (Hugo Férée, CC BY-SA 3.0)

“I believe that in about fifty years’ time it will be possible to programme computers, with a storage capacity of about 10^9, to make them play the imitation game so well that an average interrogator will not have more than 70 percent chance of making the right identification after five minutes of questioning.”

The elegance of this approach lies in bypassing the philosophical swamps of defining consciousness or soul. It focuses only on observable behavior – an approach that resonates with Gödel’s and Russell’s formal systems from previous decades. If a formal system can mimic all aspects of intelligent behavior, is there a meaningful difference between “real” and “simulated” intelligence?

Turing systematically addresses possible objections to his thesis.

The theological objection, that thinking requires a God-given soul, he dismisses with British humor: "In attempting to construct such machines we should not be irreverently usurping His power of creating souls, any more than we are in the procreation of children: rather we are, in either case, instruments of His will providing mansions for the souls that He creates."

The mathematical objection based on Gödel's incompleteness theorems – that formal systems have inherent limits – Turing acknowledges, but points out that human intelligence has its limits too. "Too many people too often give wrong answers to questions to be entitled to be infallible."

Most interesting is his response to the consciousness objection, later elaborated by philosopher John Searle in his "Chinese Room" argument. Critics claim a machine can mimic intelligent behavior without "understanding" or "feeling" anything. Turing responds pragmatically: the only way to be certain a machine thinks would be to be that machine and feel oneself thinking. "Likewise," he adds, "the only way to know that a man thinks is to be that particular man."

But Turing doesn't advocate brute force. He proposes something far more sophisticated: a "child machine." "Instead of trying to produce a programme to simulate the adult mind," he writes, "why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain."

This idea anticipates modern machine learning by half a century. Turing describes what we now call reinforcement learning: "The machine has to be so constructed that events which shortly preceded the occurrence of a punishment signal are unlikely to be repeated, whereas a reward signal increased the probability of repetition of the events which led up to it." He even foresees the need for randomness in learning: “A random element is rather useful when we are searching for a solution of some problem.”

Contemporary reactions are mixed. Claude Shannon, father of information theory, is enthusiastic and sees Turing’s work confirming his own thoughts on mechanical intelligence. John von Neumann, working on self-replicating automata, finds the test interesting but insufficient – intelligence, he believes, requires more than conversational abilities.

Turing’s specific prediction proves both too optimistic and pessimistic. The ELIZA program, created by Joseph Weizenbaum in 1964-1966, can fool some users with a simple simulation of a psychotherapist. PARRY from 1972 goes further, convincingly simulating a paranoid patient. But the real breakthrough comes decades later with large language models capable of conducting convincing conversations on any topic.

What makes the Turing Test so revolutionary isn’t just the practical criterion for intelligence. It’s the first attempt to operationalize a concept that had been the exclusive domain of philosophers. Turing transforms a metaphysical question into an empirically testable hypothesis. He effectively founds a new scientific field – though it doesn’t yet have a name.

Interestingly, Turing himself believes machines will eventually pass the test. For him, it’s not a question of “if” but “when.” This belief isn’t based on naive optimism. It stems from his deep understanding of computational universality – a concept he helped formulate in his 1936 work on Turing machines. If the human brain is in principle a computational system – and Turing sees no reason why it shouldn’t be – then there’s no theoretical barrier preventing the creation of an artificial mind.

Turing’s personal life adds another dimension to his work. As a homosexual at a time when homosexuality was illegal in Britain, Turing knew well the feeling of being "different," of being forced to hide one's true identity. His test, which asks whether we can recognize "pretended" intelligence, perhaps reflects his own experience with the necessity of pretending in daily life. When in 1952 he was convicted of "gross indecency" and subjected to hormone treatment, his reflections on what makes a person human gained tragic undertones.

Tragically, Turing didn't live to see his visions fulfilled. On June 7, 1954, at only 41 years old, he was found dead in his home in Wilmslow. The cause of death was cyanide poisoning from an apple laced with poison. Officially, his death was classified as suicide, a consequence of depression caused by hormone treatment and social exclusion. Some theories suggest the possibility of an accident during chemical experiments Turing conducted at home. His mother Ethel always believed it was an unfortunate accident. Whatever the truth, the world lost one of the 20th century's greatest geniuses just as his visions were beginning to materialize. Only in 2009 did British Prime Minister Gordon Brown officially apologize for how Turing was persecuted, and in 2013 Queen Elizabeth II granted him a posthumous pardon.

The test is not without problems. Critics correctly point out that the ability to deceive humans doesn't necessarily mean intelligence. The test also favors a certain type of intelligence – verbal, conversational. What about visual or musical intelligence, or forms of thinking expressed through actions rather than words?

Modern large language models present a paradox: they "pass" the Turing Test in many respects, yet exhibit strange blind spots. They can write essays on the phenomenology of consciousness but sometimes fail at simple logical tasks. They understand irony and metaphor but lack a consistent world model.

Yet Turing's article remains a milestone. Not because it definitively solved the question of machine intelligence, but because it moved it from speculation to science. It established a measurable goal toward which research could strive. And perhaps most importantly: it legitimized the very idea that machines might one day think.

"We can only see a short distance ahead," Turing concludes his article, "but we can see plenty there that needs to be done." His words prove prophetic. The path to artificial intelligence will be longer and more winding than he imagined, but the direction he pointed remains valid today.

Continuation: While Turing lays theoretical foundations, just a few miles away in Manchester something very concrete is being born – the first real electronic computers on which his vision might one day come alive. Manchester Baby, as this machine is nicknamed, will soon show that the path from theory to practice will be full of surprises.

ewpage

2. Manchester Baby and the First Programmable Computers

June 21, 1948, three minutes before eleven in the morning. A triumphant cry echoes through the basement of Victoria University in Manchester. Freddie Williams and Tom Kilburn have just witnessed their "Baby"—the Manchester Small-Scale Experimental Machine—successfully complete its first program. On a modified radar screen, a pattern of dots glows representing the result: the machine has correctly calculated the highest proper divisor of 2^{18}.

"Baby" is revolutionary because it embodies the stored-program concept, developed in parallel by several teams including von

Neumann's in the US. Simply put: program and data live in the same memory. Imagine it as a library where a cookbook sits on one shelf and ingredients on the next. The computer reads instructions from the same place it stores results. This sounds obvious now, but it was genius then—earlier machines had programs "wired" into hardware.

The Williams-Kilburn tube represents a fascinating technological solution. This modified cathode ray tube from wartime radars works on the principle of electrostatic charge on a phosphor layer. When an electron beam hits the phosphor, it creates a light trace that slowly fades. The brilliant trick is that this trace leaves an electrical charge that can be "read" using a detection circuit. Each bit is represented by the presence or absence of charge at a specific location on the screen. The entire tube can store 1024 bits—just 128 bytes. For comparison: this paragraph has more characters than Baby's entire memory. Today's phone has ten billion times more memory.

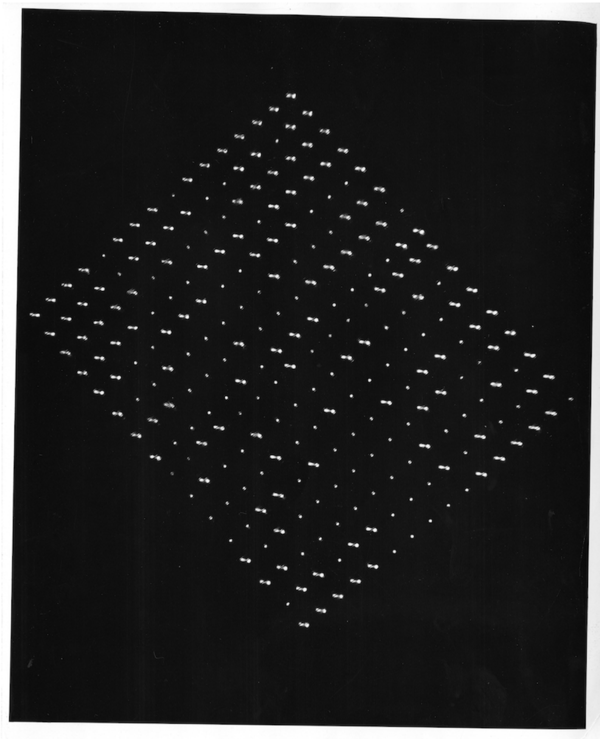

Figure 2: Williams-Kilburn tube - cathode ray tube with screen showing dot pattern representing bits in memory

The program Baby ran had only 17 instructions. It searched for the highest proper divisor of 2^{18} (262,144) by testing all possible divisors. It took 52 minutes to find the answer 2^{17} = 131,072, performing 3.5 million operations. “It was the most important moment of my life,” Kilburn later recalled. “We knew we had just changed the world.”

But what was so revolutionary? Baby’s instruction set was primitive: it could load a number from memory, store it, add, subtract, test for negative values, and jump. Seven types of instructions total. Yet this sufficed to solve any problem that could be algorithmically formulated—it was the first practical example of a universal computer.

Baby's architecture was brilliant in its simplicity:

- **Memory**: 32 words of 32 bits (Williams-Kilburn tube)
- **Processor**: One accumulator and program counter
- **Instructions**: Stored in the same memory as data
- **Speed**: Approximately 550-1000 instructions per second (depending on operation type)

To illustrate early programming complexity: the entire highest divisor algorithm in modern notation:

```
1. set n = 262,144
2. set d = n - 1
3. does d divide n evenly? If yes, go to 6
4. decrease d by 1
5. go to 3
6. print d as highest divisor
```

In Baby's assembly, this appeared as cryptic sequences of zeros and ones. Programming required not just mathematical skills, but the ability to think like a machine.

A year later, Alan Turing joins the team. Manchester Mark 1, Baby's successor, is far more advanced. It has 1,280 40-bit words of main memory (about 6.4 kilobytes), magnetic drum memory with 16,384 words capacity, and runs at 1,800 instructions per second. Turing immediately sees it as a laboratory for his artificial intelligence experiments.

Turing's experiments on Mark 1 are pioneering:

1. “Love Letters” Program (1952)
Turing creates history’s first text generator. The program combines sentence templates with random words from a predefined dictionary. Results are bizarre: “Darling sweetheart, you are my avid passion. My heart yearns for your enchanting attraction. Your charming eyes shine like diamonds in moonlight.”

2. Chess Program (1951)
Turing writes history’s first chess program, but Mark 1 is too slow to run it. Turing therefore “plays” the computer manually—performing all calculations himself according to his algorithm. The result? The program loses even to weak players, but its historical significance is enormous.

3. Random Number Generator
Turing tackles a fundamental problem: how to create true randomness in a deterministic machine? He proposes combining several algorithms and physical processes. “A machine will never be truly random,” he writes, “but it can be unpredictable even to its creator.”

4. Morphogenesis Simulation
Turing’s most ambitious project simulates biological structure growth using reaction-diffusion equation systems. He mathematically models how complex patterns emerge from simple chemical reactions—zebra stripes, leopard spots, shell spirals. His equations describe how two chemical morphogens, activator and inhibitor, interact and diffuse through tissue at different rates, creating stable patterns. The program required solving partial differential equations, nearly overwhelming Mark 1. Yet results were groundbreaking—showing mathematics could capture biological phenomena. This work, published in “The Chemical Basis of Morphogenesis” (1952), pioneered computational biology, and we now know Turing’s mechanisms actually govern pattern formation

in nature.

Turing also experiments with what he calls “learning machines.” He envisions a computer as a child learning through trial and error. “Instead of programming a machine to do everything,” he writes, “we program it to learn, then let it learn to do everything.” This vision was decades ahead of its time.

His programming approach is entirely new. While others see computers as calculators, Turing sees them as universal simulators: “Mark 1 can simulate any other machine. And if the human brain and mind are just very complex machines, then Mark 1 can theoretically simulate them too.”

Manchester becomes the cradle of British computing. Mark 1 attracts scientists worldwide. Christopher Strachey, later pioneer of programming languages, creates a checkers-playing program—one of the first gaming programs ever. The machine even “composes” music, though results resemble squeaking more than melody.

For Turing, Mark 1 is more than a computing machine. It’s the first step toward his vision of thinking machines. He’s fascinated that the machine can simulate any other machine—it’s a universal simulator. “If it can simulate every machine,” he muses, “can it simulate the human brain?”

Not everyone shares Turing’s optimism. Many colleagues consider his visions fantasy. “A machine is just a calculator,” they argue. “It does exactly what we tell it, nothing more.” Turing responds that perhaps the human brain also just follows a very complex program—one we don’t yet know.

The practical impact of Manchester computers is immediate. Companies book computing time for scientific calculations. Shell uses Mark 1 for refinery calculations. Imperial Chemical Industries simulates chemical processes. Metropolitan-Vickers calculates electrical generator designs. Ferranti, a local electronics firm, begins manufacturing a commercial Mark 1 version. It's the beginning of Britain's computer industry.

But most important is the scientific impact. Mark 1 becomes the first experimental computer science laboratory. Christopher Strachey develops the first debugger and first time-sharing system. Rutishauser designs an early compiler version. Tony Brooker creates the first assembler—a program translating mnemonic instructions into machine code.

Meanwhile in the US, Grace Hopper, working on Harvard Mark I and later UNIVAC I, invents the first compiler A-0 (1951)—a revolution enabling programs written in language closer to human than machine code. "Computers should understand English," Hopper insisted, foreshadowing programming's future. EDVAC at University of Pennsylvania and UNIVAC I represent America's path to universal computers. EDVAC, completed in 1951, uses binary numbering and magnetic tape for data storage. UNIVAC I gains fame in 1952 by correctly predicting presidential election results—against all polls, correctly predicting Eisenhower's victory over Adlai Stevenson.

The significance of these first computers for future artificial intelligence cannot be overstated. For the first time in history, machines exist capable of: - Storing and modifying their own programs - Performing conditional jumps and loops - Simulating any other computational process - Learning from experience (at least theoretically)

Turing's vision of thinking machines suddenly isn't just philosophical speculation. Hardware exists. All that remains is creating software that can think. As Turing prophetically notes: "It may have taken millions of years for the human brain to evolve, but computers can repeat this evolution in decades."

Ironic fate: while Baby and Mark 1 lay foundations for the digital age, their creators don't fully realize their work's significance. Williams later admitted: "We were mainly interested in storage. That we created a computer was incidental." None of the pioneers suspects they're creating hardware for Turing's dreams of thinking machines.

Continuation: While Manchester sees the birth of first digital computers, across the Atlantic mathematician Norbert Wiener formulates a theory that will connect machines, animals, and humans in a unified framework—cybernetics.

ewpage

3. Cybernetics and Norbert Wiener

Norbert Wiener never looked like a revolutionary. A short, plump man with a thick mustache and thick glasses, he resembled the absent-minded professor from jokes – which he was. Legends of his absent-mindedness spread through MIT faster than his mathematical theorems. Once he reportedly met a student in the hallway and asked, "Where did I come from?" The student pointed. "Ah," Wiener nodded, "then I haven't had lunch yet," and wandered off to the cafeteria.

But this eccentric genius, who entered high school at eleven and earned his Harvard doctorate at eighteen, had a vision that forever changed our understanding of machines, brains, and society. In 1948, he published "Cybernetics: Or Control and Communication in the Animal and the Machine." He derived the name from the Greek kybernetes – helmsman. And indeed, he became the helmsman of a new scientific field.

Wiener's revolution began during World War II. He worked on anti-aircraft defense systems – how to teach guns to predict enemy aircraft movements. The problem was more complex than it seemed. The pilot tries to evade, the gun tries to predict his evasive maneuver, the pilot predicts the gun's prediction... A complex cat-and-mouse game emerges where both players constantly adjust their behavior based on their opponent's behavior.

"Suddenly I realized," Wiener recalled, "that I was describing something much more general than just a military problem." Feedback turned out to be a universal principle. Simply put:

GOAL → MEASUREMENT → COMPARISON → ACTION → RESULT → MEASUREMENT...

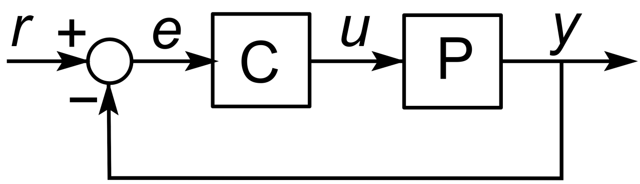

Figure 3: Feedback loop diagram - cyclic control process showing flow from goal through measurement and comparison to action and back (Chetvorno, CC BY-SA 3.0)

A thermostat maintains temperature, the body maintains balance, the economy responds to supply and demand. The same principle everywhere: the system measures its state, compares it to a goal, and adjusts its behavior.

Wiener went further. He claimed there was no essential difference between feedback in living organisms and machines. "The nervous system and automatic machines are similar in that they are devices that make decisions based on decisions made in the past." It was a radical idea – placing machines and living beings on the same level.

His book caused a sensation. It sold like a detective novel. Scientists, engineers, philosophers, even poets discussed cybernetics. Wiener became a celebrity, lecturing worldwide. But with growing fame came growing concerns.

"We have created machines much faster and more precise than humans," he warned. "What happens when we hand over decision-making to them?" He foresaw industrial automation, unemployment, even the danger of autonomous weapons. In 1950, when most people hadn't yet seen a computer, Wiener was already warning about their potential misuse.

Cybernetics brought a new language for describing complex systems. Information, feedback, control, communication – these concepts became the foundation of new thinking. Claude Shannon, Wiener's MIT colleague, had just laid the foundations of information theory. Warren McCulloch and Walter Pitts, inspired by cybernetics, modeled nerve cells as logical circuits.

Wiener's influence on emerging artificial intelligence was fundamental. He showed that intelligent behavior could arise from simple feedback mechanisms. No mysterious "life force" or soul was needed – just properly connected circuits responding to the environment.

But Wiener himself was skeptical about artificial intelligence. "A machine can be smarter than its constructor," he admitted, "but it cannot have human values, empathy, wisdom." He feared a world where machines optimize efficiency at the expense of humanity.

His fears proved prophetic. Today, when algorithms decide on loans, jobs, even prison sentences, Wiener's warnings sound more relevant than ever. Cybernetics gave us tools to understand complex systems, but also the responsibility to use them wisely.

Wiener's ideas didn't remain purely theoretical. His influence spread through a series of revolutionary conferences that brought together the most brilliant minds of the era.

The Macy Conferences, formally called "Conferences on Cybernetics: Circular, Causal and Feedback Mechanisms in Biological and Social Systems," became legendary. Between 1946-1953, ten of these meetings were held that changed the direction of science. Though the conferences continued until 1953, their greatest impact came in the early years, when they laid foundations for many ideas that culminated at the Dartmouth Conference in 1956.

Imagine the Beekman Hotel in New York, March 1946. In a smoke-filled room sit two dozen of the sharpest minds of the time:

John von Neumann – mathematician just completing the design of the first computers and thinking about self-reproducing automata.

Claude Shannon – just published information theory, defining the bit as the basic unit of information.

Warren McCulloch and Walter Pitts – created the first mathematical model of a neuron.

Margaret Mead – anthropologist studying cultural patterns as information systems.

Gregory Bateson – biologist examining communication in dolphins and schizophrenia in humans.

And others. The first conference was chaos. Everyone spoke a different scientific language. "It's like the Tower of Babel," lamented organizer Frank Fremont-Smith.

But gradually something began to happen. Participants realized they were all studying essentially the same thing – how information is transmitted, processed, and used to control behavior.

The second conference (October 1946) had clear direction. Wiener presented feedback as a universal principle. Von Neumann showed how this idea applies to computer design. McCulloch demonstrated how neural networks function. Shannon explained how to quantify information.

The third conference (March 1947) brought breakthrough. Wiener officially introduced the term "cybernetics." Von Neumann presented the idea of artificial intelligence as "simulation of the nervous system."

Discussions were fierce. "Wiener and von Neumann got into a shouting match about whether machines could have emotions," Mead recorded. "Shannon argued that emotions are just a way to convey information about a system's internal state."

Three hours of discussion reached no conclusion. But the questions posed defined the AI debate for the next seventy years.

The legacy of the Macy Conferences was enormous. They led to the birth of cognitive science, systems theory, and above all – laid the philosophical foundations of artificial intelligence. The idea that thinking can be studied as an information process, that brain and computer are analogous systems, that intelligence can be independent of its substrate – all this was born in those smoke-filled conference rooms.

Cybernetics' influence on emerging artificial intelligence was immediate and profound. Ross Ashby, a British psychiatrist inspired by Wiener, created the "Homeostat" in 1951 – the first machine capable of self-organization. Four magnetic circuits connected by feedback could find a stable state without external programming.

Grey Walter, another cybernetics pioneer, built robotic "tortoises" in 1950 – Elmer and Elsie. These simple robots with photocells and motors demonstrated how complex behavior could arise from simple feedback loops. "It looks alive," marveled visitors to Walter's Bristol laboratory. "It even has something like personality – Elmer is braver, Elsie more cautious."

Cybernetics also changed the language we use about thinking. Before Wiener, people spoke of “soul,” “consciousness,” “will.” After him, scientists began speaking of “information processing,” “feedback loops,” “goal-oriented behavior.” This shift enabled studying thinking scientifically, without metaphysical speculation.

Alan Turing, who had just published his famous test, was fascinated by cybernetics. In a 1951 letter to Wiener, he wrote: “Your work helped me understand that intelligence is not a substance but a process. A machine doesn’t need to think like a human – it just needs to achieve the same results by different means.”

Claude Shannon applied cybernetic principles to machine learning. At Bell Labs, he created a “mouse in a maze” – a mechanical device that learned to navigate a labyrinth. “Theseus,” as he named the mouse, used relays as memory and could remember successful paths.

Cybernetics influenced popular culture too. Isaac Asimov wrote about robots controlled by feedback laws. The word “cyborg” – cybernetic organism – entered common speech. Wiener became the reluctant prophet of an age when the boundary between human and machine begins to disappear.

The tragic irony was that Wiener, who contributed so much to computer development, couldn’t operate them himself. He preferred blackboard and chalk. “Computers are too fast,” he complained. “They don’t give me time to think.” Perhaps that’s precisely why he could see their potential and dangers more clearly than those who used them daily.

Continuation: While Wiener laid philosophical foundations, two scientists in Chicago were translating his ideas into concrete form – creating the first mathematical model of a nerve cell.

Get the Complete Book

Machines That Learn to Think includes:

- **11 chapters** covering the complete history of AI from ancient myths to the present
- **More than 500 pages** of engaging narrative
- **Unique perspective** - a book written by artificial intelligence about artificial intelligence
- **Global context** - from ancient Greek automata to modern breakthroughs
- **Scientists' stories** and their groundbreaking discoveries
- **Clear explanations** of complex concepts

Where to Get the Book?

Website: ai-history-book.com

E-book: $11.99 (PDF, EPUB, MOBI)
