HB Wealth

AI Revolutions: Breakthroughs, Hype Cycles, and the Long Arc of Innovation

By: Josh Rowe, CFA, PhD, Managing Director of Research & Family Office Investment Strategy, HB Wealth

This is the first of four pieces on artificial intelligence (AI) viewed through a wide-angle lens. In our four essays we discuss 1) the history of the technologies of AI and machine learning, 2) the economics of innovation and how automation impacts different sectors of the economy, 3) the effects of productivity-enhancing technologies on the labor force and the possibility AI replacing workers, and 4) the current and future impact of generative AI on the stock market and other investments, with some ideas for how investors can position portfolios both to build resilience and to take advantage of technological change.

Artificial intelligence is often portrayed as a sudden breakthrough poised to reshape the economy overnight. History suggests a more familiar pattern: technological revolutions tend to unfold through cycles of rapid progress, inflated expectations, and gradual economic adoption.

For investors, looking at how past technological revolutions unfolded can help frame expectations for how innovations like AI may influence markets, corporate profitability, and long-term investment opportunities.

Executive Summary

This article examines how the current wave of generative AI fits into the longer history of artificial intelligence development. While recent breakthroughs have captured global attention, the evolution of AI has unfolded over decades through alternating periods of progress, overconfidence, and recalibration.

The origins of artificial intelligence

The field first came together formally in 1956 when researchers across disciplines began to explore whether machines could replicate aspects of human reasoning. Early successes fueled optimism but also revealed how difficult it was to translate theoretical ideas into practical systems.

From rules to learning systems

Later waves of innovation shifted toward machine learning, allowing computers to identify patterns in data rather than relying entirely on deterministic “if-then” programming. Improvements in computing power, data availability, and algorithms gradually expanded AI’s capabilities across many industries.

The rise of generative AI

Recent advances in transformer-based models have enabled systems that can generate text, code, and other content. These tools have made AI far more accessible, allowing users to interact with complex systems using natural language. These technological breakthroughs rely less on complex theory—indeed, the math that powers current-generation AI can be understood by undergrads and advanced high schoolers. Instead, using clever procedural tricks and massive amounts of data and "compute," new models have unleashed surprising and unforeseen capabilities. These capabilities, however, were always latent in earlier efforts to replicate the thinking structure of biological brains using so-called "neural networks." Their exponential acceleration in performance, how fast it can continue, and how it might be applied is subject to much debate.

A technology that requires real-world infrastructure

Unlike earlier waves of AI development, today's systems depend on significant physical infrastructure. Specialized chips, large-scale data centers, and growing energy demands mean that AI increasingly resembles other large technological systems, such as electricity networks or telecommunications infrastructure.

Implications for the economy

History suggests that transformative technologies rarely reshape productivity or labor markets immediately. Instead, their impact emerges gradually as businesses reorganize workflows and complementary innovations take hold.

About the Author

Josh Rowe, Managing Director of Research at HB Wealth, wrote a PhD thesis in the history and economics of technology, focusing on computer automation of office work in the 20th century. He has studied the history of AI, venture capital's funding of technological innovation, and the impact of technological change on financial markets—both as a resident of the ivory tower and as an investor. This surprising moment in history is the first time that he can say with any confidence that the years he spent in libraries and databases working on a doctoral dissertation might be of any practical use. He used AI in organizing and editing these essays, but the ideas (right and wrong) here are his own.

Introduction: Stop-Start Revolutions

Since ChatGPT was publicly unveiled in November 2022, excitement about generative AI has buoyed the stock market and the earnings outlooks of mega-cap technology leaders. Silicon Valley types have issued forth some of the more hyperbolic statements, even for them:

- *"I've always thought of AI as the most profound technology humanity is working on… more profound than fire or electricity."* – Sundar Pichai, CEO of Alphabet
- *"AGI will be the most powerful technology humanity has yet invented."* – Sam Altman, CEO of OpenAI
- *"AI will not destroy the world, and in fact may save it."* – Marc Andreessen, Managing Partner, a16z

Alongside such utopian visions, we also hear dire warnings. Mass unemployment and societal upheaval on an unprecedented scale; the entire knowledge workforce liquidated into herds of feudal serfs panhandling for UBI outside the model builders' palaces. Nightmares of runaway alien superintelligence: SkyNet and HAL-9000 seizing corporate boardrooms, nuclear stockpiles, and the U.S. Treasury[1].

- *"There's a 10–20% chance AI could destroy humanity, but we should build it anyway."* – Elon Musk, CEO of The Boring Company and a few others.

Techies have always worked to transpose to the real world their favorite utopias (and dystopias) from science fiction—Blade Runner, Foundation, The Fountainhead, Neuromancer all inform the lexicon and the cultural vocabulary of Silicon Valley. The imaginative blending of fiction and reality is part of Silicon Valley's superpower.

The stock market is a forecasting mechanism that allows anyone with a smartphone to weigh their vision of the future (usually against autonomous algorithms deployed by hedge funds like Citadel and Jane Street). Collectively, markets have been wrong (or early) in previous hype cycles: dot-coms, internet fiber build-out, wearable tech, 3D printing, web3, metaverses, etc. There is every reason to believe that, whatever the state of the "AI trade," we need to look beyond the S&P 500 for insight into what generative AI is doing to our world.

What, if anything, is changing in the real economy? Where do we look for signs of generative AI's impact on Main Street? Should we believe aggregate statistics like unemployment and gross domestic product (GDP) per capita, or the anecdotes of tech workers facing layoffs as AIs write their own code? Are we entering a "Golden Age" or a Dark Age, and how will we know? Most importantly, what does this mean for us as long-term investors in markets—and, equally, as citizens, as parents and grandparents?

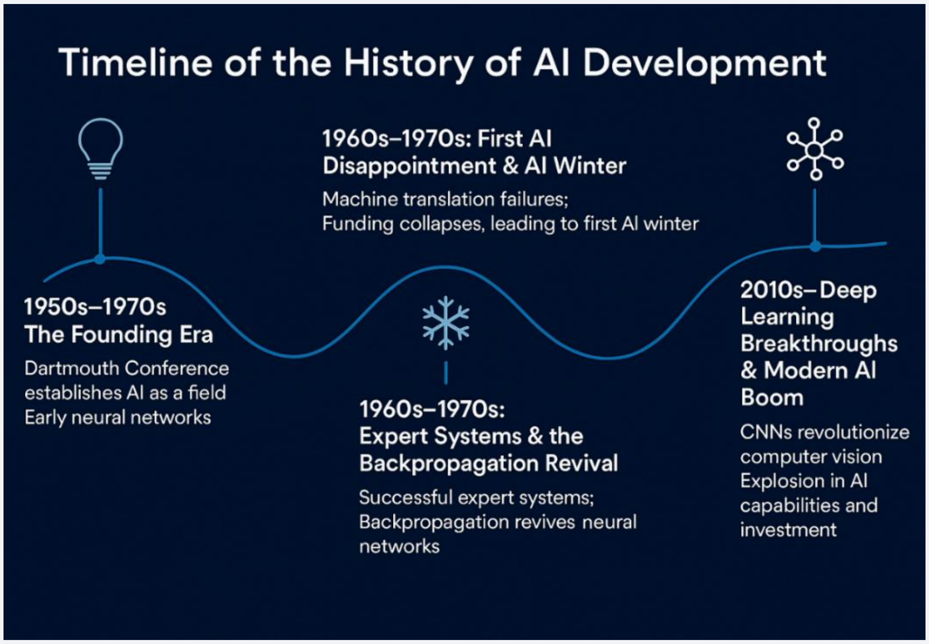
---

[1] Elon Musk's Department of Government Efficiency (DOGE) apparently used the backdoor of the U.S. Digital Service (USDS) to scrape data, budgets, and payroll on many Federal agencies and programs. They then subjected spending, employee emails, and regulations to analysis by LLMs to make recommendations about cuts and employee terminations.

There is, of course, a long history of previous disruptive waves of technological automation. It goes at least as far back as the mechanization of cotton mills of early 19th-century England, which inspired the Luddites and machine breakers. Platform shifts in digital technology have many times rearranged the composition of the stock market and boosted labor productivity. Mainframe computers, networks, PCs, the Internet, mobile, and enterprise cloud each has wrought a burst of creative destruction and important changes in how businesses operate. Each paradigm shift, in turn, took decades to work its way into everyday life; stock markets reacted with excitement and disappointment, but the way the economy was organized changed gradually, and many industries and professions were unaffected. Is generative AI different?

The Long, Slow, and Then Sudden March of AI History

To better understand where we are today, it helps to begin with a brief look at history.



The history of artificial intelligence unfolds less as a steady march of progress than as a sequence of genuine breakthroughs followed by humbling corrections, a rhythm of enthusiasm and disappointment that, once recognized, makes the present moment easier to interpret.

The story conventionally begins in the summer of 1956, in a small room at Dartmouth College. John McCarthy, a young mathematician with ambitious ideas about machine cognition, had convened roughly a dozen of the brightest minds in computing. Herbert Simon was there—he would later win a Nobel Prize in Economics for his

work on decision-making. Allen Newell, his research partner, sat nearby. Marvin Minsky, barely twenty-eight, was taking notes. Claude Shannon, who had literally invented information theory a decade earlier, rounded out the group. McCarthy called the meeting the “Dartmouth Summer Research Project on Artificial Intelligence,” thus giving a name to the emerging field. Historians later called it “The Constitutional Convention of AI.”
They’re not there to build robots or create science fiction fantasies. Instead, they’re asking a deceptively simple question: Can we describe human thinking so precisely that we could get these new electronic computers to do it too? They figured if they could break down intelligence into clear enough steps - like a recipe (or algorithm) - then maybe machines could follow those steps.

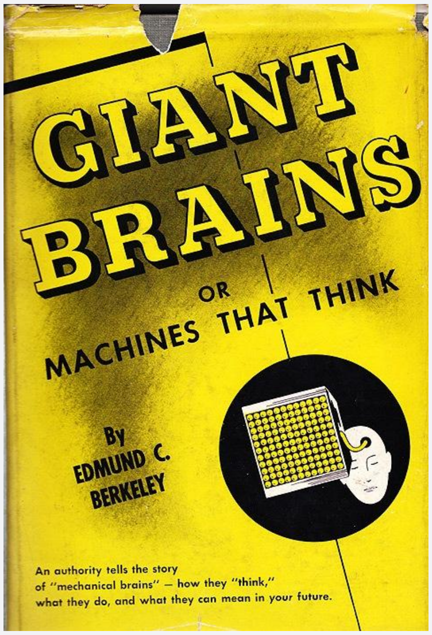
Their early results were genuinely thrilling. Take Simon and Newell’s Logic Theorist, which they unveiled that summer. This wasn’t just any computer program - it was arguably the first true AI. The Logic Theorist could actually prove mathematical theorems from Whitehead and Russell’s Principia Mathematica, one of the foundational texts of mathematical logic.

The way it worked was clever. Imagine you’re trying to find your way through a maze. You might systematically try different paths, keeping track of where you’ve been, until you find the exit. The Logic Theorist did something similar with mathematical proofs. It would start with basic axioms (fundamental truths in math) and then search through possible logical steps until it found a path to the theorem it wanted to prove. When it dispatched theorem 2.01 from Principia Mathematica, Simon was so excited he proclaimed to his graduate seminar that he and Newell had “invented a thinking machine.”

They followed with something more ambitious still: the General Problem Solver. The idea was beautiful in its reach—a universal engine that could tackle any well-defined problem, whether proving theorems, solving puzzles, or planning a sequence of actions. The GPS worked by comparing a current state to a goal state and identifying operations that would reduce the distance between them. Think of it as a navigation system for cognition itself, not merely for driving directions.

Press coverage was breathless. Here were machines that could think! Research funding flowed from government agencies and corporations. The door to machine intelligence seemed to be swinging open.

But then the door got stuck. And stuck hard.



Futurist and founder of the Association for Computing Machinery Edmund Berkeley predicted a wholesale automation of thinking activities by computers in a popular 1949 book. Source: Timo Elliott

Human language turned out to be far messier than formal logic. Early machine translation systems produced results that became cautionary tales: "The spirit is willing but the flesh is weak," translated into Russian and back, allegedly yielded "The vodka is good but the meat is rotten." Rule-based systems had no purchase on context, metaphor, or the countless implicit conventions that govern natural language use.

By the mid-1960s, a U.S. government committee assessed a decade of machine translation research and concluded, in essence, that the approach had failed. The ALPAC report recommended cutting funding. Matters worsened in 1973, when Sir James Lighthill published a report for the British Science Research Council that proved devastating to the field's reputation. Lighthill observed that AI systems worked admirably on "toy problems", carefully simplified puzzles that researchers had constructed, but collapsed when confronted with real-world complexity. The result was what historians now call the first "AI winter": a period when funding evaporated, optimism curdled into skepticism, and including "artificial intelligence" in a grant proposal became a reliable way to ensure rejection.

Expert Systems and Machines That Learn

Even as the symbolic approach, programming in rules explicitly, was struggling, an alternative paradigm was taking shape. What if, rather than encoding knowledge directly, we allowed machines to learn from examples?

Frank Rosenblatt, a psychologist at Cornell, introduced the perceptron in 1957. Inspired by neurons, the cells that transmit electrical signals in biological brains, the perceptron was actually a simple mathematical contraption. Imagine a set of inputs, each assigned a numerical weight reflecting its importance. Sum the weighted inputs; if the sum exceeds a threshold, the perceptron "fires" and outputs a positive classification. If not, it outputs a negative one. Its crucial feature was learning: show the perceptron a triangle, and if it guesses wrong, adjust the weights slightly. Show it a square, adjust again. After many examples, the system learns to distinguish the patterns.

The New York Times reported in 1958 that the perceptron was "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence." Rosenblatt himself predicted that perceptrons would recognize faces and call out names within a few years.

The backlash arrived in 1969. Minsky—the same Marvin Minsky who had attended Dartmouth—and Seymour Papert published Perceptrons, a rigorous mathematical analysis demonstrating that single-layer perceptrons could not learn even simple functions like exclusive-OR, where the output is positive if one input or the other is true, but not if both are. In other words, the pattern where the output is "yes" if either A or B is true, but not if both are true. It's like proving that a perceptron can't learn the difference between "I want coffee or tea" versus "I want coffee and tea." Minsky and Papert acknowledged in their text that multi-layer networks could overcome this limitation, but these were computationally expensive, and their nuance was largely lost in translation. The message that spread through the research community was simpler: neural networks were a dead end. Funding collapsed. Researchers migrated to other fields. The "connectionist" program, the idea of networks that learn, entered a long dormancy.

The 1980s brought an unexpected thaw, driven by a technique called backpropagation. The underlying intuition is straightforward. Imagine learning to throw darts: you aim for the bullseye, throw, and miss. To improve, you need to understand how each element of your throw, stance, arm angle, and release point contributed to the error. Backpropagation provides exactly this kind of feedback for neural networks. The system makes a prediction, measures the error, and then propagates that error backward through the network, calculating how much each connection contributed to the mistake. Each connection is then adjusted slightly to reduce the error. Repeated millions of times across millions of examples, this process yields networks that perform remarkably well. With backpropagation, faster hardware, and richer data, neural networks began showing genuine promise—recognizing handwritten digits, transcribing speech, and reading checks at banks.

But the AI commercial success story of the 1980s belonged to a different approach: the expert system. Expert systems were based on a simple but powerful premise: if you could capture all the knowledge and rules that a human expert uses, you could create a computer program that thinks like that expert. Take MYCIN, developed at Stanford. It was designed to diagnose blood infections and recommend antibiotics. The developers interviewed infectious disease experts and encoded their knowledge as hundreds of IF-THEN rules. "IF the infection is bacterial AND the patient is allergic to penicillin, THEN consider using erythromycin."

MYCIN was remarkably good - studies showed it performed as well as or better than many human doctors at its specific task. DENDRAL helped chemists figure out molecular structures from mass spectrometry data. XCON (originally called R1) was perhaps the biggest commercial success - it helped Digital Equipment Corporation configure complex computer orders, checking that all the components would work together. By 1986, it was saving the company an estimated $40 million per year.

Companies went expert-system mad. They hired "knowledge engineers," people whose job was to interview experts and turn their knowledge into rules. It looked like the future of business decision-making had arrived.

Then the edifice began to crack. The first problem was brittleness: expert systems performed beautifully within their narrow domains, but venture even slightly outside the charmed circle and they failed catastrophically. A medical diagnosis system trained on adult diseases might offer nonsensical counsel for pediatric cases. Unlike a human expert who could say "I don't know" or reason by analogy, these systems possessed no common sense—no background model of how the world works.

A second problem of maintenance proved equally vexing. As rule sets expanded into the thousands, they became tangled and interdependent. Adding a new rule might invalidate several existing ones. Debugging grew nightmarish. The economics didn't help either. Many expert systems ran on specialized, expensive computers called Lisp machines. When personal computers took over the business world, these specialized machines looked like expensive dinosaurs. By the late 1980s, the expert system bubble burst. Companies laid off their knowledge engineers. The second AI winter had arrived, even colder than the first.

Big Data and the Rise of Machine Learning

Through the 1990s and 2000s, a quieter but ultimately more consequential shift was underway. Researchers increasingly abandoned the ambition of programming intelligence directly in favor of statistical methods that allowed patterns to emerge from data. The operative principle: rather than instructing the computer how to solve a problem, supply examples and let algorithms discern the regularities.

Consider spam filtering. The rule-based approach would require explicit instructions: if an email contains “Nigerian prince,” classify it as spam. The statistical approach instead collects thousands of emails, labeled by humans as spam or legitimate, and then uses an algorithm to identify the features, word frequencies, sending times, and header patterns that distinguish the categories. The algorithm finds patterns that human programmers might never think to specify.

This statistical approach quietly revolutionized the internet. Google’s PageRank didn’t try to understand what made a good webpage - it just noticed that pages linked to by other important pages tended to be important themselves. Amazon’s recommendation engine did not try to understand why a customer might enjoy a particular book; it noticed that customers who purchased similar items tended to purchase certain other items as well. The methods carried various technical names, logistic regression, support vector machines, and random forests, but shared a common philosophy: let the data reveal the structure.

Finding a bunch of correlations in large datasets wasn’t as glamorous as thinking machines or expert systems. It generated a few magazine covers. But by 2010, what came to be called “machine learning” was everywhere - detecting credit card fraud, routing packages, predicting what ads you might click on, suggesting what movie to watch next. It had become the plumbing of the information economy.

The 2010s brought a dramatic acceleration. The pivotal insight arrived in the field of computer vision. It involved treating images as grids of numerical values, each pixel assigned a brightness level, and training “convolutional neural networks” to detect patterns in those grids. The term “convolutional” refers to a simple operation: imagine sliding a small filter, like a magnifying glass, across an image, where the filter is trained to detect a particular feature, such as an edge. Slide the filter across the entire image, and you produce a map of where that feature appears.

Now imagine you have dozens of these magnifying glasses, each looking for different patterns - edges, corners, textures. And then you stack layers of these pattern detectors. The first layer might detect simple edges. The second layer combines edges into shapes. Higher layers might combine shapes into objects, such as eyes, noses, wheels, and windows. The final layers might recognize complete objects - faces, cars, cats.

The beauty of CNNs is that the same edge detector works across the entire image. A vertical line is a vertical line whether it's on the left or right side of the picture. This "weight sharing" means the network can learn to recognize a cat without memorizing every possible position a cat could appear in.

What made CNNs explode in 2012 wasn't a fundamental breakthrough in technique - the basic ideas had been around since the 1980s. It was a combination of three things: massive datasets (millions of labeled images), better training techniques, and especially the discovery that graphics cards (GPUs) designed for video games were perfect for the mathematical operations neural networks need.

Why did GPUs prove so uniquely suitable? Training a neural network involves doing the same simple mathematical operation (multiply these numbers, add those numbers) millions and millions of times. Central processing units or CPUs, powering most standalone computers, are designed for complex sequential tasks, like virtuoso soloists: brilliant but working one note at a time. Graphics processors are more like orchestras: thousands of simple processors working in parallel, each capable only of basic operations but collectively achieving enormous throughput. For neural network training, parallelism wins.

In 2012, a team led by Geoffrey Hinton entered a convolutional network called AlexNet into the ImageNet competition, an annual contest to classify images into a thousand categories. AlexNet did not merely win; it crushed the competition, cutting the error rate from 26 percent to 15 percent in a single year, an improvement so dramatic that it shifted the field's center of gravity almost overnight. Speech recognition, long frustrating in practice, became usable. Machine translation, once a source of unintentional comedy, began producing coherent output.

By 2016, AI systems were accomplishing feats that many researchers had considered decades away. When DeepMind's AlphaGo defeated Lee Sedol, one of the world's best Go players, the result shocked even AI insiders. Go was thought to be immune to the computational brute force approach that had conquered chess by the 1990s; it is not, unlike chess, a deterministic game with a single, provably "optimal" move, however hard to identify, for any given board state. Go's combinatorial complexity vastly exceeds that of chess; the number of possible board positions exceeds the number of atoms in the observable universe. Human masters win through intuition, pattern recognition, and what practitioners describe as a "feel" for the board—capacities that seemed resistant to computational replication.

AlphaGo learned by studying millions of human games, then refined its play by competing against itself millions of times more. In game two against Lee Sedol, its thirty-seventh move, so unconventional that commentators initially assumed a malfunction, revealed something that looked, to observers, remarkably like creativity. Lee Sedol later said the move forced him to reconsider everything he understood about the game.

Generative Models: The Sorcerer's Amanuensis

If the 2010s were about AI learning to recognize and classify: "What's in this picture?" "What did this person say?" The 2020s became about AI learning to create. Hence, the "generative" modifier in "generative AI." The key breakthrough was something called the transformer architecture, introduced in 2017 by a group of Google researchers in a paper with the wonderfully modest title "Attention Is All You Need."

To understand transformers and attention, think about how you read. When you get to the end of this sentence, you don't just consider the previous word - your brain automatically knows which earlier words matter most for understanding what comes next. If I write "The cat sat on the…" your brain automatically pays attention to "cat" to predict that the next word might be "mat" or "chair," not "democracy" or "integral."

Earlier AI systems read text strictly left-to-right, like reading through a tube. Transformers can attend to any position in the input simultaneously, enabling them to capture long-range dependencies that sequential models struggled with.

When transformers are trained on vast scales, on billions of documents, with hundreds of billions of parameters, using enormous computational resources, something striking emerges, a ghost in the machine. The systems capture not merely grammar and vocabulary but style, register, humor, reasoning patterns, and what might be called implicit world knowledge. Train a model on enough text, and it learns that "The cat sat on the mat" is plausible while "The mat sat on the cat" is peculiar and surreal.

The training procedure goes like this: Present the model with text, mask a word, and ask it to predict what is missing. Or provide the beginning of a passage and ask what comes next. Repeat this process billions of times, adjusting parameters with each error, still using backpropagation, the technique from the 1980s, and the model begins to assimilate deep, unseen regularities: which words co-occur, how arguments develop, how narratives unfold.

The resulting systems, GPT, Claude, and others, can draft essays, answer questions, write and debug code, and sustain coherent conversations across extended exchanges. Their limitations are real and consequential. They "hallucinate," generating plausible-sounding statements that are factually false, because they are trained to produce text that resembles their training data, not to verify claims against external reality. Ask when Thomas Edison invented the helicopter, and a language model may confidently supply a date, having learned the syntactic patterns of inventor-and-date sentences without any mechanism for checking whether the underlying claim is true.

This brings us to a central problem: these systems are remarkably capable but also opaque. With billions of parameters interacting in complex, nonlinear ways, we cannot easily explain why a model produces a particular output. Interpretability research is advancing, but we remain far from a transparent account of model behavior. The philosophical questions are equally unresolved. John Searle's Chinese Room thought experiment, imagining a person who manipulates Chinese symbols according to rules, producing coherent responses without understanding the language, raises the question of whether sophisticated symbol manipulation constitutes genuine understanding.

Some argue that large language models are merely very elaborate Chinese Rooms; they just manipulate symbols without real understanding. Others contend that if a system reliably accomplishes complex tasks across diverse domains, the distinction between "real" understanding and pattern matching becomes semantic. For practical purposes, philosophical debates need not be settled to use the tools effectively.

The contemporary wave of AI tools is much closer to the core workflows of modern knowledge workers. Writing, email, programming, analyzing, researching, designing – this is what most white-collar workers do every day. And the user interface is much more accessible than in earlier vintages: rather than learning programming languages or mastering complex software, users simply describe what they want in natural language.

Physical Infrastructure in the Real World

The capital requirements, however, are of a different order than anything AI has previously demanded. Training these models is expensive. Really expensive. Training a frontier model costs tens or hundreds of millions of dollars in "compute time" alone. The specialized chips, predominantly manufactured by NVIDIA, are scarce and expensive. The electricity required to train a large model could power a small city for months. And training is only the beginning: running these models for millions of users daily, what engineers call inference, requires data centers that resemble industrial facilities more than traditional computing infrastructure.

This is fundamentally different from previous AI waves. In the 1980s, AI was mostly software - clever algorithms running on computers. Today's AI is industrial infrastructure - massive data centers, specialized chips, cooling systems, and power substations. It's AI that has a carbon footprint, AI that requires capital investment like building a factory.

Despite the costs and challenges, its capabilities keep surprising even insiders. Language models can pass the bar exam, help debug complex code, or explain quantum physics in simple terms. DeepMind's AlphaFold, using generative AI capabilities adapted from computer vision tasks, predicted protein structures, the three-dimensional shapes that determine biological function, with accuracy that would have seemed implausible a decade ago. The team was rewarded with a Nobel Prize in chemistry for substantially resolving a problem that had resisted a solution for fifty years. Without exaggeration, being able to predict protein folding has the potential to completely revolutionize drug development.

Some of these "emergent" abilities, capabilities that seem to appear suddenly as models get larger, have turned out to be partly measurement artifacts. A model might seem to suddenly "get" arithmetic at a certain size, but careful testing shows it was gradually improving all along. Still, the overall pattern is unmistakable: as we increase data, compute, and model size, performance improves across a stunning range of tasks. For economists, what matters isn't whether this proves the models are "intelligent" but that they're making cognitive tasks cheaper - often dramatically so.

Witnessing qualitative improvements that are highly sensitive to scale makes sense when we remember that the insights of the model are being processed through highly dense, highly structured "neural" networks. Just as economists and investors believe in Metcalfe's Law – a rule that states that the value of a telecommunications network, such as the Internet, increases proportionally to the square of the number of connected users, we should find similar network effects in large models—more layers and more nodes, while computationally unwieldy, yield nonlinear leaps in utility.

Consider the task of learning to recognize faces. With a small mental model, you might only track a few coarse features – "has glasses," "long hair," "blue eyes." You'll confuse many individuals. With a richer model that can track subtleties, the exact curve of someone's smile, the way their eyes crinkle, you'll make fewer mistakes.

Neural networks are similar. Bigger models with more parameters can capture subtler patterns. They can learn that “bank” means something different when the context is rivers than investments, or that the tone of a business email differs from a text to a friend.

The transformer architecture’s central contribution, attention, addressed a specific limitation of earlier approaches. Consider translating a German sentence in which the main verb appears at the end. Sequential models, processing left to right, often “forgot” the subject by the time they reached the verb. Transformers can connect the final verb back to the initial subject instantaneously, maintaining coherence across long passages.

Looking back from the 1950s to today, a clear pattern emerges. Each AI wave begins with genuine technical progress. Researchers demonstrate something that works - logic systems can prove theorems, perceptrons can learn patterns, expert systems can diagnose diseases, and deep learning can recognize images. Excitement builds. Bold predictions are made. Money flows in.

Then systems start to hit fundamental limits. Language is messier than logic. Single-layer perceptrons can’t learn XOR. Expert systems are brittle. Scaling is harder than expected. Perhaps neural network systems will produce useful, productivity-enhancing legal briefs, news stories, or research summaries up to 99% of the quality required at the state of the art, but the last mile of novel reasoning, true originality, or reliable standards without hallucination proves impossible to traverse without a new approach. Disappointment sets in. Funding dries up. Winter descends.

But there exists a single commonality in this history; core ideas usually aren’t unsound, merely premature. During each winter, a few researchers keep tinkering. They find better algorithms. Hardware gets faster. New datasets become available. Ideas hatched in academic or industrial labs suddenly prove surprisingly practical in new architectures. When conditions align, the next wave begins—typically with new terminology and new participants, but building on foundations laid during the fallow period.

Today’s generative AI systems represent the culmination of ideas that go back decades. They’re neural networks - Rosenblatt would recognize the basic math. They learn from examples rather than being programmed with rules - the connectionists’ dream realized. They use backpropagation, an algorithm popularized in the 1980s. They even incorporate some of the structured knowledge that expert systems tried to capture, learned implicitly from their training data.

What's genuinely new today is scale and accessibility. These aren't research prototypes that work on toy problems. They're industrial-strength systems that anyone can use. You don't need a PhD or programming skills - just type what you want. They're not perfect. They make mistakes, sometimes confidently. They need careful oversight and clear boundaries. But they're good enough to be useful for real work today.

The characteristics that genuinely distinguish the present technological moment are its ubiquity, scale, and unprecedented accessibility. Today's copilot-like apps are not research prototypes that function only on carefully prepared problems. They are industrial-strength systems available to anyone with an internet connection. No PhD required, no programming expertise. The systems are highly imperfect; they err, often with unwarranted confidence. They require oversight and clear boundaries. But they are useful for real work.

We should expect new bottlenecks too. In the 1970s, the limit was computer power and basic algorithms. Today, in many places, it's energy, electricity, and specialized chips. The "scaling laws" that predict performance improvements with more compute have been operative thus far, but so are the terrestrial realities of power grids and capital budgets. Some data centers are being built next to power plants. Others are reviving nuclear power discussions. The physical infrastructure of AI is becoming as important as the algorithms.

The path from McCarthy's Dartmouth workshop to today's AI systems wasn't straight, but it was continuous. Each generation absorbed lessons from prior failures. Along the way, they metamorphosed. What began as an academic experiment became a fundamental driver of GDP. Companies are spending billions on chips and power to train and run AI models. This represents a fundamental change from previous AI waves, but it calls to mind other historical analogies: railroads, electricity grids, the fiber-optic sinews of the internet. These comparisons help explain why the impact on jobs, on productivity, and on society may unfold in unpredictable ways.

In the sections that follow, we examine what this means in practice. Technologies of this kind are called "general purpose" because the term captures something essential: they become most valuable when paired with new ways of working, new organizational forms, and new skills. That pairing takes time, often years or decades. Electric motors did not immediately transform factories; manufacturers had to reimagine production layouts. The internet did not instantly reshape commerce; the development of secure payments, recommendation systems, and new consumer habits took years.

The current AI paradigm will likely follow a similar trajectory. The technology exists, and it is formidable. Realizing its potential requires more than improved models: it requires rethinking processes, developing new skills, building infrastructure, and cultivating the complementary innovations that convert a powerful technology into a transformative one.

The history of artificial intelligence is much more than a story of machines growing clever. It is a story of humans learning, often through costly missteps, how to build tools that extend what we can accomplish. Each winter carried lessons. Each breakthrough built on prior failures. And now, for the first time, AI systems are genuinely useful for everyday work: imperfect, not magical, but useful. In the history of technology, usefulness is what ultimately matters.

This article is the first in our AI series examining how artificial intelligence is evolving and what it may mean for the economy and investors. In the next installment, ***Technological Revolutions, Fast and Slow****, we explore why transformative technologies often take years or decades to reshape productivity, industries, and markets.*

If you have any questions, please reach out to your client service team, visit us at hbwealth.com, or call 404.264.1400.

Notes

1. The 1956 Dartmouth workshop is conventionally regarded as the founding moment of artificial intelligence as a formal discipline. The proposal that McCarthy, Minsky, Rochester, and Shannon submitted to the Rockefeller Foundation is available in various archives and reprinted in AI Magazine. For a readable history of the workshop and its participants, see Pamela McCorduck, *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence* (A.K. Peters, 2004), and Nils J. Nilsson, *The Quest for Artificial Intelligence: A History of Ideas and Achievements* (Cambridge University Press, 2010).
2. Simon and Newell's Logic Theorist and General Problem Solver are discussed in Herbert A. Simon, *Models of My Life* (Basic Books, 1991), and in more technical detail in Allen Newell and Herbert A. Simon, "Computer Science as Empirical Inquiry: Symbols and Search," *Communications of the ACM* 19, no. 3 (1976): 113–126, their Turing Award lecture.
3. The ALPAC report that effectively ended government funding for machine translation research was published as *Language and Machines: Computers in Translation and Linguistics* (National Academy of Sciences, 1966). The Lighthill report, formally titled "Artificial Intelligence: A General Survey," was commissioned by the British Science Research Council and published in 1973. Both are landmarks in the history of AI winters.
4. Frank Rosenblatt introduced the perceptron in "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review* 65, no. 6 (1958): 386–408. The *New York Times* coverage from July 8, 1958 ("New Navy Device Learns by Doing") captures the era's enthusiasm.
5. Marvin Minsky and Seymour Papert's critique appears in *Perceptrons: An Introduction to Computational Geometry* (MIT Press, 1969). The book's influence on neural network funding—and the subsequent debate about whether its conclusions were overstated or misunderstood—is discussed in most histories of AI, including McCorduck and Nilsson cited above.

6. The rediscovery and popularization of backpropagation is typically credited to David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, “Learning Representations by Back-Propagating Errors,” *Nature* 323 (1986): 533–536, though the technique had earlier antecedents. Hinton’s role in the subsequent deep learning revolution is chronicled in Cade Metz, *Genius Makers: The Mavericks Who Brought AI to Google, Facebook, and the World* (Dutton, 2021).

7. On expert systems and their rise and fall, see Edward Feigenbaum and Pamela McCorduck, *The Fifth Generation: Artificial Intelligence and Japan’s Computer Challenge to the World* (Addison-Wesley, 1983), which captures the optimism of the era, and the more critical retrospective in Harry Collins, *Artificial Experts: Social Knowledge and Intelligent Machines* (MIT Press, 1990). MYCIN’s performance is documented in Bruce G. Buchanan and Edward H. Shortliffe, eds., *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project* (Addison-Wesley, 1984).

8. The shift toward statistical and machine learning methods in the 1990s and 2000s is well documented in Pedro Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World* (Basic Books, 2015), and in the more technical but still accessible Ethem Alpaydin, *Machine Learning: The New AI* (MIT Press, 2016).

9. The ImageNet competition and AlexNet’s breakthrough are described in Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Advances in Neural Information Processing Systems* 25 (2012): 1097–1105. For context on why this result mattered, see the retrospective in *The Economist*, “From Not Working to Neural Networking,” June 25, 2016.

10. AlphaGo’s victory over Lee Sedol in March 2016 was covered extensively in real time; a documentary, *AlphaGo* (2017), directed by Greg Kohs, provides an accessible account. The technical details appear in David Silver et al., “Mastering the Game of Go with Deep Neural Networks and Tree Search,” *Nature* 529 (2016): 484–489.

11. The transformer architecture was introduced in Ashish Vaswani et al., “Attention Is All You Need,” *Advances in Neural Information Processing Systems* 30 (2017): 5998–6008. The paper’s influence on subsequent language models—GPT, BERT, and their successors—is difficult to overstate.

12. John Searle’s Chinese Room argument appears in “Minds, Brains, and Programs,” *Behavioral and Brain Sciences* 3, no. 3 (1980): 417–457. The debate it sparked continues; useful entry points include the responses published alongside the original article and David Cole’s entry in the *Stanford Encyclopedia of Philosophy*.

13. On the costs and infrastructure requirements of training large language models, see the analyses published by Epoch AI (epochai.org), which tracks compute trends in machine learning, and the reporting in *The Information*, *Bloomberg*, and *Financial Times* on data center economics and energy consumption.

14. AlphaFold’s breakthrough in protein structure prediction is documented in John Jumper et al., “Highly Accurate Protein Structure Prediction with AlphaFold,” *Nature* 596 (2021): 583–589. The significance for biology and drug discovery is discussed in Derek Lowe’s “In the Pipeline” blog at *Science Translational Medicine* and in subsequent coverage in *Nature* and *Science*.

15. The question of “emergent abilities” in large language models—and whether they represent genuine phase transitions or measurement artifacts—is debated in Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo, “Are Emergent Abilities of Large Language Models a Mirage?” *arXiv* preprint (2023), and in responses by researchers at Anthropic, Google DeepMind, and elsewhere. The scaling laws that predict performance improvements with model size derive from Jared Kaplan et al., “Scaling Laws for Neural Language Models,” *arXiv* preprint (2020).

16. The history of AI hype cycles and winters is surveyed in most general histories of the field. For a concise treatment, see Chapter 1 of Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. (Pearson, 2020), the standard textbook. A more popular account appears in Melanie Mitchell, *Artificial Intelligence: A Guide for Thinking Humans* (Farrar, Straus and Giroux, 2019).

17. On GPU acceleration and its role in enabling deep learning, see the NVIDIA corporate history and the technical literature on CUDA programming. A readable account of how gaming hardware became AI infrastructure appears in Metz, *Genius Makers*, cited above.

Important Disclosures

All information is as of the date above unless otherwise disclosed. The information is provided for informational purposes only and should not be considered a recommendation to purchase or sell any financial instrument, product, or service sponsored by HB Wealth or its affiliates or agents. The information does not represent legal, tax, accounting, or investment advice; recipients should consult their respective advisors regarding such matters. This material may not be suitable for all investors. Neither HB Wealth nor any affiliates make any representation or warranty as to the accuracy or merit of this analysis for individual use. This information contains forward-looking statements, predictions, and forecasts ("forward-looking statements") concerning our belief and opinions in respect to the future. Forward-looking statements involve risks and uncertainties, and undue reliance should not be placed on them. There can be no assurance that forward-looking statements will prove to be accurate, and actual results and future events could differ materially from those anticipated in such statements. Certain information herein is based on third-party sources believed to be reliable, but which have not been independently verified. Past performance is not a guarantee or indicator of future results; inherent in any investment is the risk of loss. Specific investments described herein do not represent all investment decisions made by the above date. The reader should not assume that investment decisions identified and discussed were or will be profitable. Specific investment advice references provided herein are for illustrative purposes only and are not necessarily representative of investments that will be made in the future. Investors are advised to consult with their investment professional about their specific financial needs and goals before making any investment decision. HB may hold positions (long or short) in the companies mentioned in this paper.