AGENTIC AI EXPLAINED

From Neural Networks to Autonomous Agents: How It Works, How to Build It, and What It Means

MICHAEL FRASER

Agentic AI Explained

From Neural Networks to Autonomous Agents: How It Works, How to Build It, and What It Means

Steve Publications

This book is available at https://leanpub.com/agenticaiexplained

This version was published on 2026-09-20

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

From Neural Networks to Autonomous Agents: How It Works, How to Build It, and What It Means

You have heard the term “agentic AI” everywhere. News headlines promise robots that work for you. Product managers ask their teams to build “an agent” for everything. Vendors sell you autonomy, and sometimes you are not sure whether to believe them. This book explains what agentic AI actually is, how it works under the hood, what it can and cannot do reliably, how real systems are designed and built, and what risks and trade-offs you should understand before trusting one with anything important. It assumes you know nothing about artificial intelligence and walks you from the ground up, with patience, concrete examples, and no hype. By the time you finish, you will be able to cut through marketing language, think clearly about where agents can help, and understand what people mean when they say something is or is not ready for production.

Introduction

In early 2025, a product manager at a mid-sized software company received this message from her team lead: "Can we build an AI agent to handle the entire onboarding flow for new enterprise customers?" She nodded, added it to the roadmap, and asked her engineering lead how long it would take. The engineering lead did not answer immediately. He had seen agents in demos. He had also watched them fail in unpredictable ways when given open-ended tasks. He knew the answer depended on what they actually meant by "handle the entire onboarding flow."

That moment of uncertainty is the starting point for this book.

Agentic AI has moved from academic papers and conference slides into boardrooms, product roadmaps, and sales conversations. The term itself is not new (artificial intelligence researchers have been talking about "agents" since the 1990s), but the capabilities behind it have changed dramatically in the last three years. Modern AI agents can read email, browse the web, call APIs, write code, query databases, coordinate with other agents, and execute multi-step workflows with a degree of autonomy that was unimaginable a decade ago. They are not science fiction. They are real systems running in production right now, and they are improving rapidly.

At the same time, they are deeply imperfect. They hallucinate. They lose track of context. They make confident mistakes. They can be tricked by hidden instructions in web pages they are told to read. They can misuse their tools. They can be expensive and slow. They can drift quietly over time, degrading in performance without anyone noticing until something goes wrong. Understanding what is real and what is hype (what agents can do today, what is coming soon, and what remains genuinely hard) requires understanding how these systems actually work.

This book is designed to give you that understanding from the ground up.

What this book covers

We start at the very beginning. Before you can understand what an AI agent is, you need to know what artificial intelligence means, how machines learn

from data, and why neural networks are the foundation of everything that follows. The early chapters explain the history of AI in broad strokes, covering the decades of research that led to today's breakthroughs. You will learn how machine learning algorithms discover patterns, how neural networks transform raw data into representations that support intelligent behavior, and how a specific architectural innovation called the transformer made modern AI possible.

Then we move to large language models, the technology that makes today's AI feel so different from anything before it. You will learn what a language model actually is at its core (a system that predicts the next piece of text) and why this seemingly simple task produces something capable of writing code, answering questions, translating languages, and explaining complex topics. We will look inside the model to understand how it processes your prompts, how tokens and context windows work, how the attention mechanism lets it track relationships between ideas, and why its outputs feel surprisingly human.

We will also talk honestly about what language models are not. They do not "know" facts in the way humans know facts. They do not "understand" the world. They do not have beliefs, goals, or intentions. They are incredibly sophisticated statistical pattern machines, and they are best and worst at very specific kinds of things. Understanding these limitations is critical, because many of the failures of AI agents trace back to misunderstandings about the underlying models.

After you understand language models, we will explain what makes an agent different from a chatbot. A chatbot takes a prompt and returns a response. An agent takes a goal and pursues it, using tools, memory, and planning to execute multi-step tasks over time. We will break down the anatomy of a modern AI agent, explaining each component: the model that reasons, the memory that retains information across interactions, the tools that let it interact with the real world, the planner that breaks down goals into steps, and the orchestrator that coordinates everything.

You will learn about specific architectures and patterns used in real agentic systems, including the ReAct framework that interleaves reasoning with action, retrieval-augmented generation that grounds agents in real data, reflection loops that let agents critique and fix their own work, and multi-agent systems where specialized agents collaborate on complex tasks. We will examine frameworks like LangGraph, CrewAI, and AutoGen that developers use to build these systems, not in code-heavy detail but in conceptual terms that clarify

how they fit together.

We will also cover what can go wrong, extensively. Agents are not just chatbots that make typos. They can send wrong emails, execute bad code, delete data, leak sensitive information, or fall into infinite loops that burn through your API budget. We will discuss prompt injection attacks, where hidden instructions trick agents into doing things they should not do. We will talk about guardrails, permissions, and human oversight patterns that keep agents safe in production. We will examine evaluation frameworks, observability tools, and monitoring practices that help teams detect and fix problems.

Finally, we will look at where agents are actually useful today. We will survey real-world applications across software development, research, customer service, business operations, healthcare, finance, and education, separating proven use cases from experimental ones. We will discuss the limitations that still constrain what agents can do reliably, including problems with long-horizon planning, context management, and consistency. We will examine the broader implications of agentic AI, including effects on jobs, the concentration of power in a few technology companies, the environmental cost of training and running massive models, and the alignment problem of ensuring powerful systems do what we actually want.

How to read this book

This book is written for intelligent non-specialists. It assumes you can think carefully and follow a logical argument, but it does not assume you have a degree in computer science, statistics, or mathematics. Technical terms are defined when first introduced. Complex concepts are explained with analogies and concrete examples. Code appears only sparingly, as illustrative snippets rather than tutorials. The goal is conceptual clarity, not the ability to implement everything from scratch.

If you are technically curious, you will find the material goes deeper than most popular explanations. You will learn about the mechanics of attention, the geometry of embeddings, the structure of planning algorithms, the architecture of multi-agent coordination, and the specifics of production safety patterns. If you want to build agents, this book will give you the vocabulary and mental models you need to read documentation, evaluate frameworks, and

design systems. If you are a manager, product leader, or executive, you will gain enough technical fluency to ask good questions, assess claims, and make informed decisions about when and how to use agentic AI.

Read the book in order. Each chapter builds on the previous ones. You can skip ahead if you already know the foundations, but the progression is intentional: from intelligence to learning, from learning to neural networks, from neural networks to transformers, from transformers to language models, from language models to agents, from single agents to multi-agent systems, from capabilities to limitations to applications to implications.

You do not need to memorize anything. Read slowly enough to understand. When a concept feels opaque, reread the explanation or try the analogy a second time. The book will make more sense on the second pass, which is normal.

Now let us begin at the beginning. What even is intelligence, and what does it mean to build a machine that has some of it?

Chapter 1: What Is Intelligence, Anyway?

An Agent at Work

Imagine a system that does the following, all in one session, with minimal human intervention:

You ask it to prepare a competitive analysis report on cloud storage providers. It starts by searching the web for recent pricing announcements and feature updates. It reads several vendor websites and blog posts. It retrieves your company's internal notes on current storage costs from a knowledge base. It writes a first draft of the report. It notices gaps, searches for more information, and revises the draft. It formats the final document, generates charts from data it collected, saves it to a shared drive, and sends a summary email to the stakeholders you specified. It took about fifteen minutes.

This is not science fiction. This is a system that could be built today using existing AI technologies, and systems like this are being deployed by forward-looking organizations. It is not perfect. It might miss something important, or get a fact wrong, or format a chart badly. But it is doing real work, autonomously, across multiple tools and systems.

This is what people mean when they talk about agentic AI.

Before we understand how such a system works, we need to understand what we are trying to build. What is intelligence? What does it mean to make something artificial that exhibits it? And how did we get here?

Defining Intelligence

Intelligence is one of those words that almost everyone understands and almost no one can define precisely. In everyday language, we call someone

intelligent if they learn quickly, adapt to new situations, solve problems creatively, or see connections others miss. In psychology, researchers debate whether intelligence is a single general ability or a collection of distinct skills. In biology, we see intelligent behavior in everything from octopuses solving puzzles to ants coordinating complex colony activities without any central controller.

In artificial intelligence, we need a working definition that is practical and testable. A common formulation in the field describes an intelligent agent as a system that:

- Perceives its environment
- Processes information to understand the current situation
- Acts in ways that achieve goals, often adapting to new or unexpected conditions

This definition is deliberately broad. It includes human beings, who perceive the world through senses, think about what to do, and act accordingly. It includes animals that navigate, hunt, and communicate. It includes thermostats, which perceive temperature and act to maintain a set point. The difference between a thermostat and a human being is not the definition of intelligence but the complexity of the environment, the difficulty of the goals, and the flexibility of the adaptation.

A crucial distinction in AI is between narrow intelligence and general intelligence. Narrow intelligence is the ability to perform well at a specific task or within a well-defined domain. A program that beats the world champion at chess is narrowly intelligent. A system that recognizes faces in photos is narrowly intelligent. These systems excel at what they are designed for but fail outside their domain. Ask the chess program to write a poem or diagnose a medical condition, and it cannot do it.

General intelligence, by contrast, is the ability to learn and apply knowledge across many different domains, much like a human being. General intelligence includes the capacity to transfer what you learned in one situation to a completely different situation, to understand novel problems without specific training on them, and to adapt flexibly when the rules change.

All artificial intelligence systems that exist today, including the most advanced ones, are narrowly intelligent. They are very good at the kinds of tasks they were trained on or can be prompted to perform, but they do not

have the broad, flexible understanding of a human being. When people talk about artificial general intelligence, or AGI, they are referring to a hypothetical system that would match or exceed human-level general intelligence. AGI does not exist yet, and there is serious disagreement among experts about whether it will ever be built, or if so, when.

The systems we will discuss in this book are advanced forms of narrow intelligence, specifically designed for language understanding and text-based reasoning, extended with tools and architectures that give them agentic capabilities. They are not AGI, even though some marketing language may suggest otherwise. Understanding this distinction is important because it clarifies both their impressive abilities and their real limitations.

The Long History of AI

Artificial intelligence, as a named field, is younger than you might think. The term was coined in 1956 at the Dartmouth Summer Research Project on Artificial Intelligence, a two-month workshop led by researchers John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon. The proposal for the workshop claimed that "every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it." This was extraordinarily optimistic. The researchers believed that human-level intelligence was just around the corner.

It was not.

The early years of AI research were dominated by symbolic AI, an approach that represented knowledge as explicit symbols and rules. Programs would manipulate these symbols using logical operations to reason about the world. In 1955, Allen Newell and Herbert Simon created the Logic Theorist, often considered the first AI program. It could prove mathematical theorems and even discovered a more elegant proof than the original for one theorem from Whitehead and Russell's Principia Mathematica. Around the same time, Arthur Samuel built a checkers-playing program that improved through experience, coining the term "machine learning" to describe its method.

These early successes fueled excitement, but also overconfidence. Researchers made bold predictions. Herbert Simon predicted in 1965 that "machines will be capable, within twenty years, of doing any work a man can do." These predictions did not come true. The symbolic approach ran

into fundamental limitations. It worked well for well-structured problems like mathematical proofs and board games, but it struggled with the messiness of the real world: perception, ambiguity, common sense reasoning, and the open-ended complexity of natural language.

By the mid-1970s, funding agencies grew skeptical. A famous 1973 report by British AI skeptic James Lighthill criticized the field for failing to deliver on its early promises. Funding in the United States and United Kingdom dropped sharply, ushering in what historians call the first AI winter, a period from roughly 1974 to 1980 when research contracts dried up and public enthusiasm collapsed.

AI survived. It recovered in the early 1980s, driven by a new technology called expert systems. These were programs that encoded the knowledge of human experts in specific domains, such as medical diagnosis or equipment troubleshooting, as sets of rules. The most famous example was XCON, developed by Digital Equipment Corporation, which configured computer systems and reportedly saved the company about forty million dollars per year by the mid-1980s. Japan's ambitious Fifth Generation Computer Project, launched in 1982, invested heavily in AI research and further stimulated interest in the West.

But expert systems had serious flaws. They were brittle, breaking when faced with situations outside their predefined rules. They were expensive to build and maintain, requiring teams of knowledge engineers to encode expert knowledge by hand. When the promised revolutionary AI did not materialize, and cheaper, more reliable traditional software proved adequate for many applications, funding dried up again. The second AI winter lasted from roughly 1987 to 1993.

Something important happened during this period. While symbolic AI dominated the public narrative, a different approach was gaining ground behind the scenes: neural networks and machine learning. Neural networks are computational models loosely inspired by the structure of the brain, consisting of interconnected nodes that adjust their connections through exposure to data. They had been studied since the 1940s, but early versions were limited by the mathematics available and the computing power of the time. A 1969 book by Marvin Minsky and Seymour Papert called Perceptrons exposed the limitations of simple neural networks, and this critique, combined with limited computational resources, caused neural network research to decline for over a decade.

By the 1980s and 1990s, researchers rediscovered and improved neural network algorithms. The key innovation was backpropagation, an efficient method for adjusting the connections in a multi-layered neural network based on its errors. This made it possible to train deeper networks that could learn more complex patterns. Machine learning gradually moved from the margins to the mainstream, quietly powering applications like email spam filters, credit scoring systems, and recommendation engines, systems so useful and ordinary that most people did not think of them as AI at all.

The real explosion came around 2012 with deep learning. Deep learning is neural network learning with many layers, hence "deep." In that year, a neural network called AlexNet, developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, won a major image recognition competition by a huge margin, reducing error rates dramatically compared to previous approaches. The victory demonstrated that deep neural networks, trained on large amounts of data with powerful graphics processing units, could solve real-world problems at an unprecedented level of performance.

Deep learning rapidly spread across domains. Neural networks became the standard for speech recognition, machine translation, image classification, and game playing. In 2016, Google DeepMind's AlphaGo defeated Lee Sedol, one of the world's best players of Go, a game considered too complex for machines. These systems were still narrowly intelligent, excelling at specific tasks. But they were learning in ways that no rule-based program ever could.

The final piece of the puzzle arrived in 2017. Google researchers published a paper titled "Attention Is All You Need," introducing a new neural network architecture called the transformer. The transformer was designed specifically for processing sequences of data, like sentences, and it introduced a mechanism called self-attention that allowed the model to weigh the importance of every part of the input relative to every other part. It was simpler and more powerful than previous sequence models and could be trained much faster using modern GPU clusters.

Transformers became the foundation of everything that followed. The GPT series of models from OpenAI (GPT-2 in 2019, GPT-3 in 2020, GPT-4 in 2023) used the transformer architecture to build increasingly capable language models, trained on enormous datasets of text from the internet. ChatGPT, released in late 2022, brought this technology to the general public in a conversational interface that felt remarkably human. Other companies followed with their own models: Claude from Anthropic, Gemini from Google,

Llama from Meta, and many others.

We are now in the era of generative AI and agentic AI, built on transformers and trained at scales that would have been inconceivable even a decade ago. These systems are not AGI. They are the most advanced form of narrow intelligence yet created, with extraordinary abilities in language understanding and generation. And when you add tools, memory, and planning on top of them, you get agents, systems that can pursue goals autonomously in ways that feel genuinely new.

Narrow AI, General AI, and the Hype Gap

The history of AI is littered with bold claims that did not pan out. The early optimists were wrong about timelines. The experts in the 1980s were wrong about expert systems being the path to machine intelligence. The AI winters were as much failures of expectation management as failures of technology.

Today's AI hype is the most intense in the field's history. Billions of dollars are flowing into AI startups and infrastructure. Major technology companies are investing hundreds of billions in AI chips, data centers, and research. Politicians, executives, and journalists talk about AI reshaping civilization. It is easy to see why the pattern of overpromising and underdelivering would make a careful reader skeptical.

Some skepticism is healthy. But there is also a real difference between today's systems and their predecessors. The capabilities of modern language models and AI agents are demonstrable, reproducible, and improving at a pace that is hard to match with any previous technology. If you talk to a modern chatbot and ask it to write code, summarize a document, explain a concept, or translate between languages, it will do these tasks at a level that would have stunned AI researchers ten years ago. The hype is often wrong about the implications and the timeline, but it is rarely wrong about the existence of the underlying capabilities.

The danger is twofold. On one side, hype inflates expectations to impossible levels. People imagine AI systems that think like humans, understand the world like humans, and can be trusted like humans. When these systems inevitably disappoint (because they do not actually think or understand or trust in human ways), the disappointment can be severe. Organizations make bad decisions,

trusting agents with tasks they are not ready for. Regulators respond with fear-driven restrictions that may slow genuinely beneficial applications. This is the cycle of hype and backlash that the field has experienced before.

On the other side, cynicism blinds us to real progress. Because AI has disappointed before, some people dismiss current advances as just another cycle, just another wave of marketing. This is also wrong. Today's systems can do things that matter. They can save time, reduce errors, automate tedious tasks, and augment human capabilities in ways that are economically and socially significant. Dismissing them entirely means missing opportunities to use them well.

The right stance is neither hype nor cynicism but careful attention to what the technology actually does, how it does it, where it works, where it fails, and at what cost. That is the stance this book takes.

Why This Book Exists

Agentic AI is powerful and confusing at the same time. The confusion is not an accident. The technology is genuinely complex, built on layers of research spanning decades. The terminology is overloaded, with words like "intelligence," "autonomous," "reasoning," and "agent" used in many different ways. The marketing is aggressive, selling visions that outpace the reality. And the pace of change is fast enough that information from even a year ago can be outdated.

This book exists to help you navigate all of that. It will give you a solid conceptual foundation, one that will help you understand new developments as they happen. It will give you the vocabulary to talk about these systems precisely. It will give you the critical thinking tools to separate real capability from marketing fiction.

Most importantly, it will give you a realistic understanding of what agentic AI can do today, what it cannot do, and what you need to watch out for if you are going to build with it, buy it, regulate it, or simply live in a world where it is everywhere.

The journey starts with learning. Before we can build agents that do things, we need to understand how machines learn in the first place.

Chapter 2: How Machines Learn

Patterns Everywhere

Everything that looks like intelligence in a machine ultimately comes down to one thing: pattern recognition.

When you recognize a face, you are matching patterns of features against patterns you have seen before. When you understand a sentence, you are recognizing patterns in word order, syntax, and semantics that your language experience has taught you. When you diagnose a problem (why your car will not start, why your code is broken, why your sales numbers dropped), you are matching the current symptoms against patterns from past experiences.

Human intelligence is pattern recognition at massive scale, layered across many domains and connected in flexible ways. Artificial intelligence is the attempt to do the same thing in a machine. The core idea of machine learning is simple: instead of writing explicit rules for how to recognize patterns, we give the machine examples of the patterns and let it figure out the rules on its own.

Consider the task of recognizing cats in photos. A rule-based approach would require you to write down all the features of a cat: pointy ears, whiskers, a certain kind of fur texture, a tail, eyes in a particular configuration. Then you would code those rules into a program. But cats come in thousands of breeds, can be photographed from any angle, in any lighting, partially hidden, or as a blurry shape in the distance. Writing rules comprehensive enough to handle all of this is practically impossible.

A machine learning approach works differently. You collect hundreds of thousands of photos, some with cats and some without, and you label them. Then you run a learning algorithm that examines the labeled photos and adjusts its internal parameters to recognize the patterns that distinguish cat photos from non-cat photos. It does not need you to tell it what a cat looks like. It figures it out from the data. The patterns it discovers are not human-readable rules but millions of numerical weights that together encode a statistical model of what a cat looks like.

This is the essence of machine learning: using data to learn patterns, rather than programming patterns by hand.

Training, Testing, and the Illusion of Understanding

Machine learning follows a basic workflow that applies across almost every application: training, testing, and deployment.

During training, you give the learning algorithm a dataset with examples. Each example has inputs and, in the most common form of learning called supervised learning, a corresponding correct output. For cat recognition, the inputs are photos and the outputs are the labels "cat" or "not cat." For spam detection, the inputs are emails and the outputs are "spam" or "not spam." For translation, the inputs are sentences in one language and the outputs are their translations.

The algorithm adjusts its internal parameters repeatedly to minimize the difference between its predictions and the correct outputs. After many passes through the data, it converges on a set of parameters that produces accurate predictions. The result is a model, which is just the collection of learned parameters ready to be used on new data.

But training on examples is not enough. A model could simply memorize the training data, repeating the correct answers for every example it has seen without actually learning anything generalizable. This is called overfitting, and we will discuss it more below. To check whether a model has genuinely learned, you test it on data it has never seen before, called the test set. If the model performs well on the test set, it means it has discovered patterns that generalize beyond the specific examples it was trained on.

This distinction between training and testing is crucial for understanding AI systems today. When a language model answers your question correctly, it is not looking up the answer. It is generalizing from the patterns it learned during training. It has never seen your exact question before, but it has seen similar questions, similar topics, and similar ways of phrasing things. It applies what it learned to produce an answer.

This generalization creates the illusion of understanding. When a language model writes an essay about climate change, or explains how a blockchain works, or debugs your code, it sounds like it understands the topic. But it does not understand in the human sense. It has learned statistical patterns

in the text about climate change, blockchains, and programming. When you ask a question, it generates text that follows those patterns in a way that is coherent, plausible, and usually helpful. It is not reasoning about the real world. It is predicting what text should come next based on what it has seen.

This distinction matters enormously. The model is good at tasks that can be solved by extending linguistic and structural patterns. It is bad at tasks that require genuine understanding of facts that are not well represented in its training data, or reasoning that goes beyond pattern extension. Many failures of AI agents trace back to confusing pattern completion with understanding.

There are three main types of machine learning: supervised, unsupervised, and reinforcement learning.

Supervised learning is what we have been describing: learning from labeled examples where the correct answer is known. It is the foundation of most AI applications you encounter daily, from recommendation systems to image classifiers.

Unsupervised learning works with unlabeled data. The algorithm tries to find structure or patterns without being told what the patterns should be. Common unsupervised tasks include clustering, where the algorithm groups similar items together, and dimensionality reduction, where the algorithm compresses high-dimensional data into a simpler representation that preserves important relationships. Language models use unsupervised learning during pretraining, where they learn from vast amounts of unlabeled text.

Reinforcement learning is learning through trial and error with feedback. An agent takes actions in an environment and receives rewards or penalties. It learns to maximize rewards by adjusting its strategy over time. This approach is how AlphaGo learned to play Go and how many game-playing AIs are trained. In modern language models, reinforcement learning with human feedback, or RLHF, is used to fine-tune models so their outputs align better with human preferences, for example, making them more helpful, less harmful, or more concise.

All three types are used in building agentic systems, but supervised and unsupervised learning form the foundation of the underlying language models.

The Magic of Gradient Descent

How does a learning algorithm actually adjust its parameters? The most common method, used for virtually all modern deep learning models, is called gradient descent.

Imagine you are standing on a mountain in thick fog. You want to reach the lowest point in the valley below, but you cannot see. What do you do? You feel the slope of the ground beneath your feet and take a step in the direction that goes downhill. Then you feel again, and take another step downhill. You repeat this process until the ground is level beneath you, meaning you have reached a local minimum.

Gradient descent works the same way, but in a space of millions or billions of dimensions. Instead of you standing on a mountain, you have a model with millions of parameters. Instead of the height of the mountain, you have a loss function that measures how wrong the model's predictions are. The algorithm feels the slope of the loss in the direction of each parameter (the gradient) and adjusts each parameter a little bit in the direction that reduces the loss.

The "learning rate" controls how big each step is. If the learning rate is too large, you might overshoot the valley and bounce around chaotically. If it is too small, you will take an impractically long time to reach the bottom. Finding the right learning rate, and sometimes adjusting it over the course of training, is a key part of training neural networks.

In practice, gradient descent is applied to batches of training data rather than individual examples, a method called stochastic gradient descent. For each batch, the algorithm computes the average gradient across all examples in the batch and updates the parameters. It repeats this for thousands or millions of batches over many training iterations. For a large language model, training might involve processing trillions of text tokens over weeks or months on thousands of GPUs.

The gradient descent process is computationally expensive. Training state-of-the-art models costs millions or tens of millions of dollars in compute and energy. This is why only a handful of organizations can train frontier models from scratch. Most other developers use existing models through APIs or by downloading pre-trained models and fine-tuning them on their own data at a much lower cost.

Backpropagation is the mathematical technique that makes gradient descent efficient for neural networks. It computes how much each parameter contributed to the overall error by working backward through the network, layer by layer. Parameters that contributed more to the error get larger updates; parameters that contributed less get smaller updates. This proportional correction is what makes neural network training scalable to models with billions of parameters.

Gradient descent is not magic. It is a straightforward optimization algorithm that has been known since the nineteenth century. But applied to neural networks with billions of parameters and trillions of data points, it produces capabilities that feel magical because the patterns it discovers are far too complex for any human to inspect or understand directly.

When Learning Fails

Machine learning systems can fail in many ways, and understanding these failure modes is critical for building reliable agentic AI. The three most fundamental issues are overfitting, underfitting, and data bias.

Overfitting occurs when a model learns the training data too well, including noise, outliers, and idiosyncrasies that do not generalize. An overfit model performs brilliantly on training examples but poorly on new data. It has memorized rather than learned. Think of a student who memorizes answers to practice questions but fails the real exam because the questions are phrased differently. Techniques to prevent overfitting include using more training data, simplifying the model, and adding regularization, penalties that discourage overly complex solutions.

Underfitting is the opposite: the model is too simple to capture the patterns in the data. It performs poorly on both training and test data because it has not learned enough. An underfit model is like a student who does not study and guesses on everything. The solution is to use a more complex model, train longer, or provide better features.

Modern deep learning models, with their billions of parameters, are typically on the side of having too much capacity, so overfitting is the more common concern. But the enormous amount of data used to train language models (trillions of tokens) means they can learn without overfitting in the traditional sense. Instead, they learn a vast and detailed representation of

the statistical patterns in the training data, which is what gives them their impressive generalization abilities.

Data bias is a different kind of failure. If the training data contains biases, the model will learn and amplify them. If most of the text about doctors in the training data refers to men, the model will associate doctors with men. If the data contains stereotypes about race, gender, nationality, or other characteristics, the model will reproduce those stereotypes. This is not because the model is prejudiced in any conscious sense. It is because the model is accurately reflecting the patterns in the data.

Bias in AI systems is a serious problem with real-world consequences. Biased hiring algorithms can discriminate against qualified candidates. Biased medical AI can provide worse recommendations for some demographic groups. Biased language models can reinforce harmful stereotypes or generate offensive content. Addressing bias requires careful curation of training data, evaluation on diverse test sets, and sometimes post-training adjustments. But it is never fully solvable, because any dataset will reflect some biases, and removing all bias can conflict with representing the world accurately.

A fourth issue, particularly relevant for language models and agentic systems, is distribution shift. Models are trained on data from a particular distribution, the mix of topics, styles, domains, and difficulty levels in the training set. When the model encounters data from a different distribution, its performance can degrade. A model trained primarily on English web text may perform poorly on domain-specific legal documents. A model trained on data from 2023 or earlier has no knowledge of events that happened after its training cutoff, and it may hallucinate information about recent events.

These failure modes are not hypothetical. They happen in production systems all the time. Building reliable AI agents requires understanding these failure modes and designing systems that detect and mitigate them.

From Rules to Learning

The shift from rule-based programming to machine learning represents one of the most important transitions in the history of computing.

Traditional software is deterministic. You write explicit rules: if this, then that. The program follows those rules exactly every time. This works beautifully for well-structured problems like financial calculations, database

queries, and protocol implementations. If you know the rules, you can encode them precisely.

But many real-world problems do not have clear rules. How do you write a rule that determines whether an email is spam? How do you write a rule that translates a sentence from French to English while preserving meaning and tone? How do you write a rule that recognizes a cat in a photo, or understands the intent behind a customer service request?

You cannot, not in any practical sense. The rules are too complex, too context-dependent, and too numerous. Machine learning solves this by reversing the process: instead of writing rules, you provide examples, and the machine discovers the rules.

This shift is fundamental to everything that follows in this book. Neural networks learn from data. Language models learn from text. AI agents learn patterns of behavior through interaction. None of them are programmed with explicit rules for how to handle every possible situation. They generalize from what they have seen.

The trade-off is that machine learning systems are probabilistic rather than deterministic. They do not give the same answer every time. They can be wrong. They can fail in unpredictable ways. They are harder to debug, because you cannot trace through explicit rules to find where something went wrong. You have to understand the statistical properties of the model and the data it was trained on.

This trade-off is worth it for problems where rule-based approaches are impractical. Language understanding, image recognition, recommendation, translation, code generation; these are all problems where machine learning has achieved levels of performance that would be impossible with hand-written rules. But it comes with responsibilities: we must design systems that are robust, transparent enough to trust, and safe enough to deploy.

With this foundation in how machines learn, we can now look at the specific architecture that makes modern AI possible: the neural network.

Chapter 3: The Brain Inspired the Network

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Biological Neurons vs. Artificial Neurons

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Layers of Abstraction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Deep Learning and the Scaling Phenomenon

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Specialized Architectures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

What Networks Actually Represent: Embeddings and the Geometry of Meaning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 4: The Transformer Revolution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Why Previous Models Hit a Wall

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Attention Explained

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Self-Attention and Context

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Encoder and Decoder

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Big Paper

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 5: What Is a Large Language Model?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Language Modeling as a Prediction Game

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Data Diet

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Size, Scale, and Emergent Abilities

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Foundation Models and the Shift in AI

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

What LLMs Are Not

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 6: Inside the LLM Brain

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Tokenization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Context Windows and Working Memory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Inference Dance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Temperature, Sampling, and Why LLMs Are Probabilistic

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Hidden Space

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 7: Prompting and the Art of Talking to LLMs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Prompt as Program

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Zero-Shot, Few-Shot, and Example-Based Prompting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chain of Thought and Structured Reasoning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

System Prompts and Role-Playing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

When Prompting Fails

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 8: Reasoning, Hallucination, and the Illusion of Understanding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Statistical Reasoning vs. Human Reasoning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chain of Thought as a Tool

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Hallucination Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Confidently Wrong

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Tools for Checking and Grounding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 9: From Chatbots to Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Chatbot Paradigm

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Agent Paradigm

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

What Makes an Agent "Agentic"

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

A Simple Example: Email Assistant vs. Email Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Spectrum of Autonomy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 10: The Anatomy of an AI Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Core Loop: Perceive, Think, Act

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Brain: The LLM as the Central Reasoning Component

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Toolbox: Tools, APIs, and How Agents Call External Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Memory System: Short-Term Context, Long-Term Storage, and Knowledge Retrieval

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Orchestrator: What Coordinates the Agent's Behavior and Enforces Constraints

Chapter 11: How Agents Use Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Function Calling Explained

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Web Search and Browsing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Code as a Tool

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Integrating with Business Systems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Tool Selection and Chaining

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 12: Memory, Knowledge, and Retrieval

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Memory Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Short-Term Memory: Conversational Context and Working Memory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Long-Term Memory: Vector Databases, Knowledge Bases, and Persistent Storage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Retrieval-Augmented Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Embeddings and Similarity Search

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 13: Planning and Decision Making

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Goal Decomposition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Planning Algorithms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Reflection and Self-Correction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Uncertainty and Adaptation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Limits of AI Planning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 14: Multi-Agent Systems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Why Multiple Agents?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Communication Protocols

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Roles and Responsibilities

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Multi-Agent Frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

When More Agents Help and When They Hurt

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 15: Building an Agentic System, Step by Step

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Problem Definition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Model Selection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Designing the Agent Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Prompt Engineering for Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Permissions, Guardrails, and Human Oversight

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 16: Guardrails, Safety, and Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Why Agents Need Guardrails

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Input and Output Filtering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Tool-Level Safety

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Human-in-the-Loop Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Monitoring and Kill Switches

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 17: Security Risks and Attack Vectors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Prompt Injection Explained

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Indirect Prompt Injection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Data Leakage and Privacy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Tool Abuse and Escalation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Attacker's View

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 18: Reliability, Evaluation, and Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Defining Success

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Evaluation Frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Red Teaming

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Observability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Continuous Improvement

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 19: Real-World Applications

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Software Development and Coding Assistants

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Research and Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Customer Service and Support

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Business Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Healthcare, Finance, and Regulated Industries

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 20: Limitations and the Realistic Horizon

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Hard Problems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Context and Attention Limits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Reliability Gap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Cost and Latency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

What's Coming Next

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 21: The Bigger Picture: Ethics, Society, and the Future

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Jobs, Automation, and Work

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Centralization and Power

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Environment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Alignment and Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Navigating the Transition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Chapter 22: Conclusion: Agentic AI, Clearly

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

What We Have Learned

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

Agentic AI in Context

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

How to Think About Agentic AI Going Forward

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

The Human Element

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/agenticaiexplained.