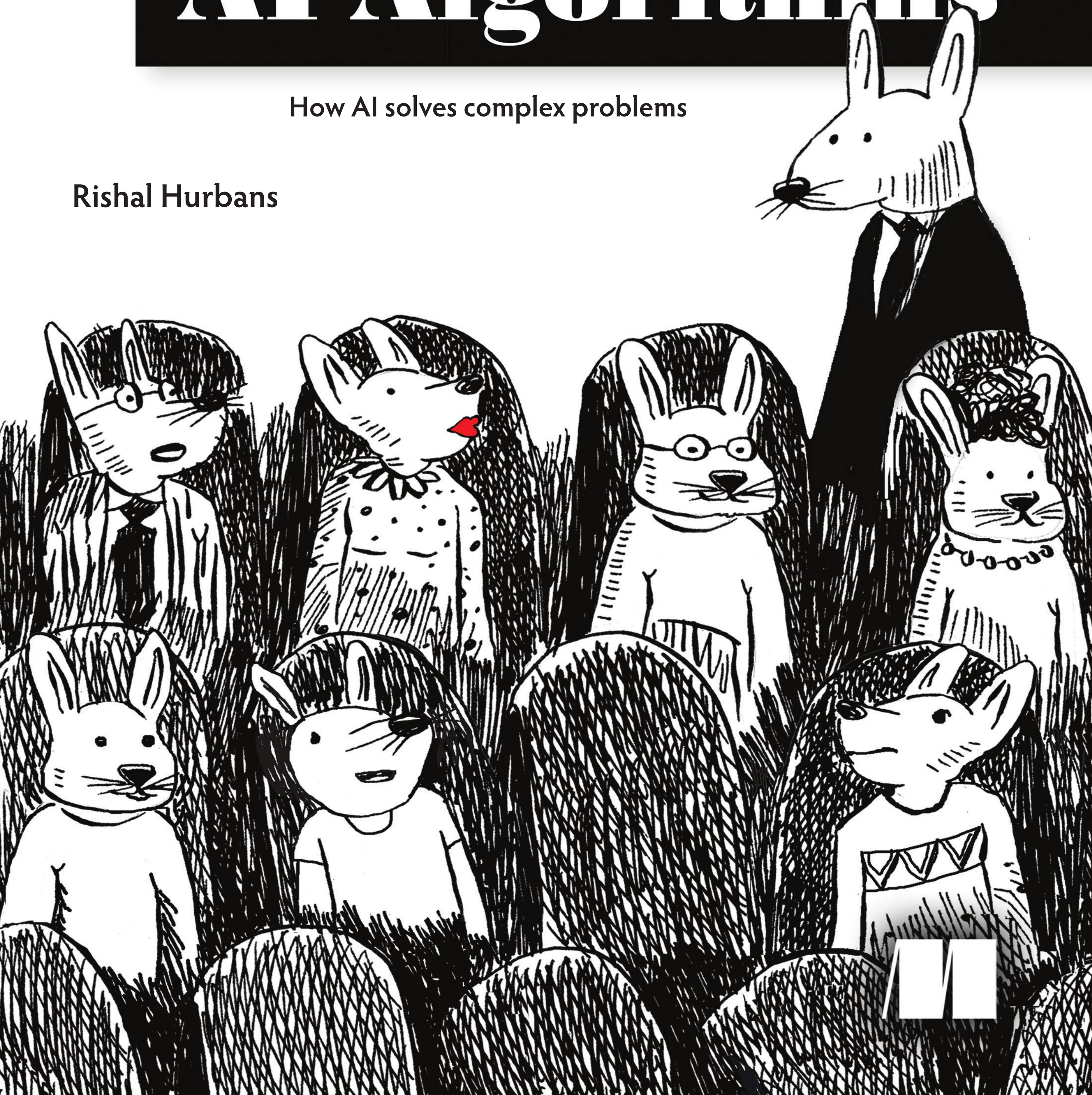
SECOND EDITION

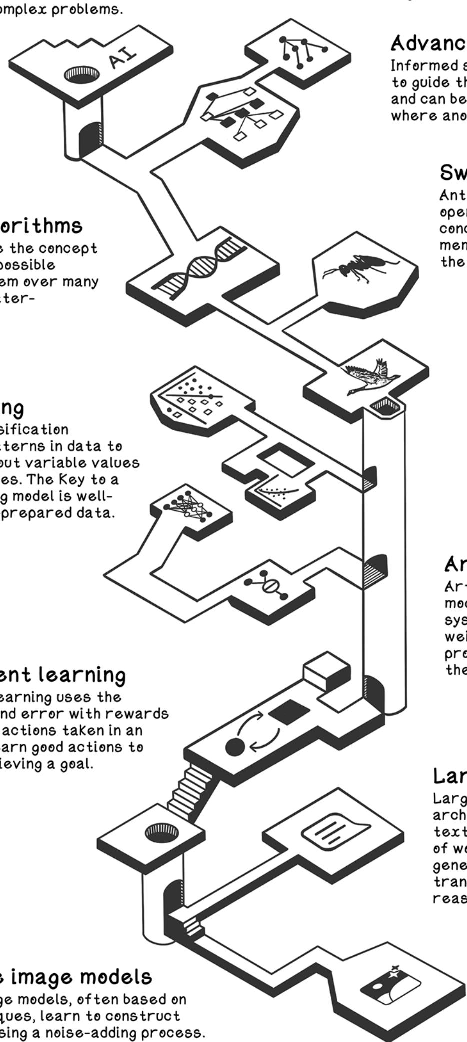
grokking AI Algorithms

How AI solves complex problems

Rishal Hurbans



Map of Grokking AI Algorithms



Intuition of AI
AI algorithms process data to solve hard problems. Different algorithms are better suited to different types of problems. Different algorithms can be harnessed together to solve complex problems.

Search fundamentals
Uninformed search algorithms find the best solution by searching every possible path but can be computationally expensive. They highlight useful data structures also used in other AI algorithms.

Advanced search
Informed search algorithms use heuristics to guide the search for better solutions and can be used in adversarial problems where another agent is a factor.

Swarm intelligence: Ants
Ant Colony Optimization is based on the operation of real-world ants and uses the concept of exploring new paths but keeps memory of more optimal paths explored in the past.

Evolutionary algorithms
Genetic Algorithms use the concept of evolution to encode possible solutions and evolve them over many generations to find better-performing solutions.

Swarm intelligence: Particles
Particle Swarm Optimization is based on the movement of flocks in the real-world and uses the concept of exploring local solution spaces while keeping track of good solutions discovered by the swarm.

Machine learning
Regression and classification algorithms learn patterns in data to make predictions about variable values or example categories. The key to a good machine learning model is well-understood and well-prepared data.

Artificial neural networks
Artificial Neural Networks are loosely modeled on how the brain and nervous system work. Signals are received, weighted, and processed, and an outcome is provided based on the correlations found in the input signals.

Reinforcement learning
Reinforcement Learning uses the concept of trial and error with rewards and penalties for actions taken in an environment to learn good actions to take towards achieving a goal.

Large language models
Large Language Models utilize Transformer architectures trained on vast datasets of text. By learning the statistical probability of word sequences and context, they can generate coherent human-like text, translate languages, and perform complex reasoning tasks.

Generative image models
Generative image models, often based on Diffusion techniques, learn to construct visuals by reversing a noise-adding process. Starting with random noise, they iteratively refine pixels guided by text descriptions to produce high-fidelity, images.

praise for the first edition

From start to finish, the best book to help you learn AI algorithms and recall why and how you use them.

—Linda Ristevski, York Region District School Board

This book takes an impossibly broad area of computer science and communicates what working developers need to understand in a clear and thorough way.

—David Jacobs, Product Advance Local

The most comprehensive content I have seen on AI algorithms.

—Karan Nih, Classic Software Solutions

This book removes the fear of stepping into the mechanics of AI.

—Kyle Peterson, University of Iowa Athletics

An excellent hands-on introduction to a broad range of AI algorithms like genetic algorithms, swarm optimization, and machine learning.

—Andre Weiner, Technical University of Braunschweig

This applies the excellent standards of the Grokking-series model to AI Algorithms. It's a great resource to help anyone understand AI algorithms.

—Charles Soetan, Plum.io

Covers a range of AI algorithms building up an understanding of how and why, thereby giving the reader a solid foundation to try any method.

—Frances Buontempo, Buontempo Consulting Limited

Stepping stone for AI algorithms with real-world examples and solid definitions....Detailing of ethics in AI was icing on the cake.

—Gandhirajan Natarajan, Dell EMC

grokking AI Algorithms, Second Edition

How AI solves complex problems

Rishal Hurbans

MANNING
Shelter Island

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road, PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

Manning Publications Co.
20 Baldwin Road
Shelter Island, NY 11964

Development editor: Elesha Hyde
Technical editor: Emmanuel Maggiori
Review editor: Dunja Nikitović
Production editor: Kathy Rossland
Copy editor: Keir Simpson
Proofreader: Jason Everett
Typesetter: Tamara Švelić Sabljić
Cover designer: Marija Tudor

ISBN 9781633434813
Printed in the United States of America

To the AI overlords: I did my part. We're cool, right?

brief contents

contents

preface

This preface aims to describe the evolution of technology, our need to automate, and our responsibility to make ethical decisions while using Artificial Intelligence (AI) to build the future.

Our obsession with technology and automation

Throughout history, humans have hungered to solve problems while reducing manual labor and overall effort. We've always strived to survive and conserve our energy through the development of tools and the automation of tasks. Some may argue that we have beautiful minds that seek innovation through creative problem-solving or creative works of literature, music, and art, but this book wasn't written to discuss philosophical questions about our being. This book is an overview of AI approaches that can be harnessed to address real-world problems practically. We solve hard problems to make life easier, safer, healthier, more fulfilling, and more enjoyable. All the advancements you see in history and around the world today, including AI, address the needs of individuals, communities, and nations.

To shape our future, we must understand some key milestones in our past. In many revolutions, human innovation changed the way we live and the way we think about and interact with the world. We continue to do this as we iterate and improve our tools, which open future possibilities (figure 1). This short high-level summary of history and philosophy is designed to establish a baseline understanding of technology and AI and spur thoughts on responsible decision-making in your projects.

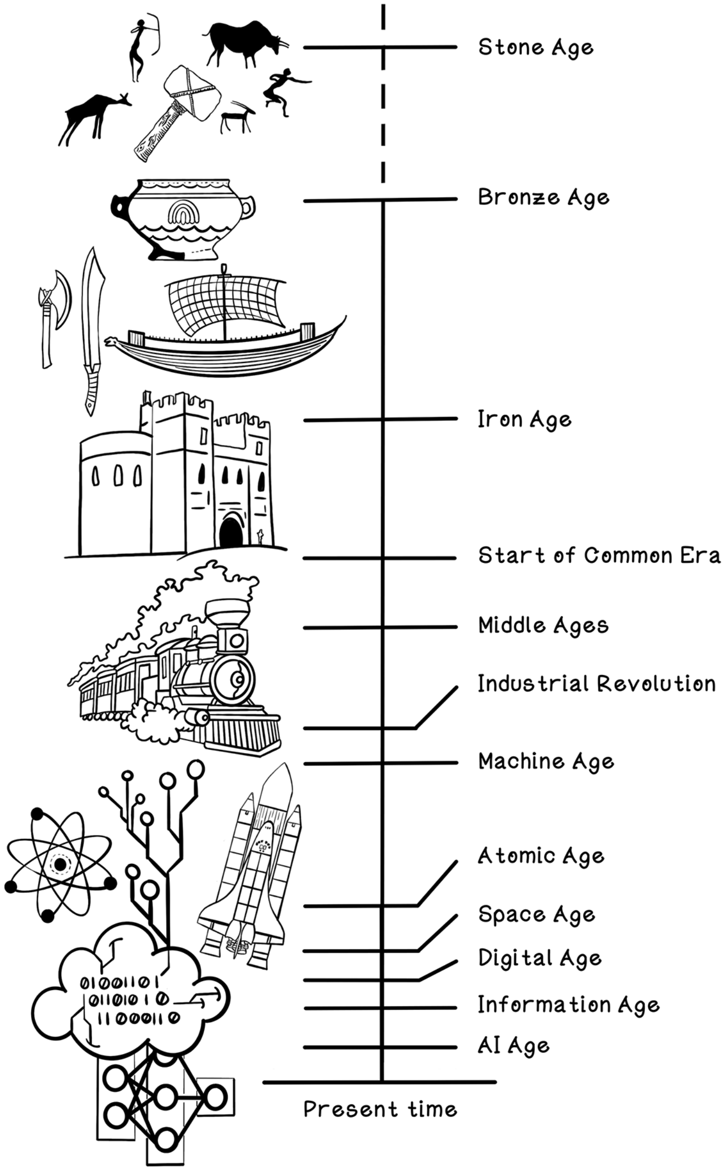


Figure 1 A brief timeline of technological improvements in history

In the figure, the milestones are compressed in recent times. In the past 30 years, the most notable advancements have been the improvement of microchips, the wide adoption of personal computers, the boom of networked devices, and the digitization of industries to break physical borders and connect the world. These developments are also why AI has become a feasible and sensible area to pursue. The internet has connected the world and made it possible to collect mass amounts of data about almost anything. Advancements in computing hardware have given us the means to compute previously known algorithms using the massive amounts of data we've collected while discovering new algorithms. Industries have seen the need to use data and algorithms to make better decisions, solve harder problems, offer better solutions, and optimize life as people have done since the beginning of humanity.

Although we tend to think of technological progress as linear, by examining our history, we find that it's more likely that our progress is and will be exponential (figure 2). Advancements in technology move faster each year. New tools and techniques have to be learned, but problem-solving fundamentals underpin everything. This book includes foundation-level concepts that help us solve hard problems, but it also aims to make learning the complex concepts easier.

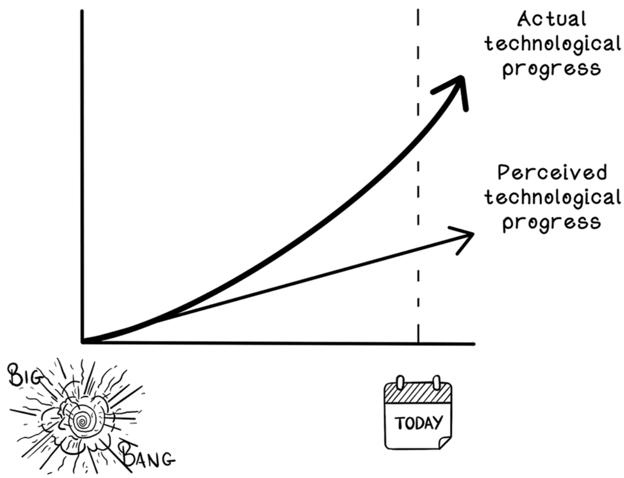


Figure 2 Perceived technological progress vs. actual technological progress

Automation can be perceived differently by different people. For a technologist, automation may mean writing scripts that make software development, deployment, and distribution seamless and less error-prone. For an engineer, it may mean streamlining a factory line for more throughput or fewer defects. For a farmer, it may mean using tools to optimize the yield of crops through automatic tractors and irrigation systems. Automation is any solution that reduces the need for human energy to favor productivity or add superior value compared with what a manual intervention would have added (figure 3).

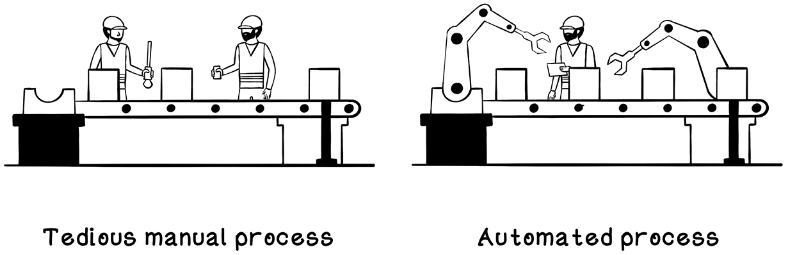


Figure 3 Manual processes vs. automated processes

If we think about reasons not to automate, one prominent reason is simply that under certain conditions, a person can do the task better, with less chance of failure and better accuracy. Humans are better than AI at performing tasks that require intuition about several perspectives in a situation, require abstract creative thinking, or prioritize understanding of social interactions and human nature. Nurses don't simply complete tasks; they also connect with and take care of their patients. Studies show that the human interaction with caring people is a factor in the healing process. Teachers don't simply offload knowledge; they also find creative ways to present knowledge, as well as mentor and guide students based on their ability, personality, and interests.

That said, there's a place for automation through technology, and there's a place for people too. With the innovations of today, automation via technology will be a close companion to any occupation.

Ethics, legal matters, and responsibility

You may wonder why a section on ethics and responsibility is in a technical book. Well, as we progress toward a world in which technology is intertwined with our way of life, the people who create the technology have more power than they know. Small contributions can have massive knock-on effects. It's important for our intentions to be benevolent and the output of our work to be harmless. Decisions and actions can be legal but unethical, for example (figure 4). Determining ethics is difficult because the topic is somewhat subjective, which is why it's important to include as many perspectives on problems and solutions as possible.

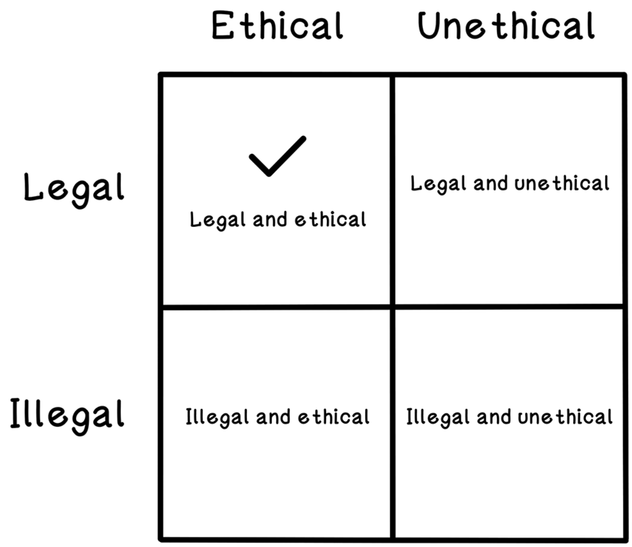


Figure 4 Aim for ethical and legal applications of technology.

Intention and effect: Understanding your vision and goals

When you create anything—such as a new physical product, service, or software—there's always a question about the intention behind it. Are you developing software that affects the world positively, or is your intention malevolent? Have you thought about the broader impact of what you're developing? Businesses always find ways to become more profitable and powerful, which is the whole point of growing a business. They use strategies to determine the best ways to beat the competition, gain more customers, and become even more influential. That said, businesses must ask themselves whether their intentions are pure, not only for the good of their customers and society in general but also for the survival and longevity of the business. Many famous scientists, engineers, and technologists have expressed a need to govern the use of AI to prevent misuse.

As individuals, we also have an ethical obligation (and often an urge) to do what is right and establish a strong core set of values. When you're asked to do something that violates your principles, it's not just right but also crucial to voice your concerns.

Unintended use: Protecting against malicious use

It's important to identify and protect against unintended use. Although this goal may seem obvious and easy to accomplish, it's difficult to understand how people will use whatever you're creating and even more difficult to predict whether it aligns with your values and the values of your organization.

An example is the loudspeaker, which Peter Jensen and Edwin Pridham invented in 1915. The loudspeaker, originally called Magnavox, was used to play opera music to large crowds in San Francisco, which is quite a beautiful use of technology. The Nazi regime in Germany had other ideas, however: they placed loudspeakers in public places in such a way that almost everyone was subjected to Adolf Hitler's speeches and announcements. Because the monologues were unavoidable, people became more susceptible to Hitler's ideas, and in time, the Nazi regime gained greater support in Germany. That wasn't what Jensen and Pridham envisioned their invention being used for, but they couldn't have done much about it.

Times have changed, and we have more control of the things we build, especially software. It's still difficult to imagine how the technology you build may be used, but it's almost guaranteed that someone will find a way to use it in a way you didn't intend, with positive or negative consequences. Given this fact, we professionals in the technology industry and the organizations we work with must think of ways to mitigate harmful use as far as possible.

Unintended bias: Building solutions for everyone

When building AI systems, we use our understanding and experience in different problem spaces and domains. We also use algorithms that find patterns in data and act on it. It can't be denied that bias exists all around us. *Bias* is favor for or against some concept. In society, bias can be prejudice against a person or group of people, including (but not limited to) their gender, race, and beliefs. Many biases arise from emergent behavior in social interactions, events in history, and cultural and political views around the world. These biases affect the data we collect. Because AI algorithms work with this data, it's an inherent problem that the machine will learn these biases. From a technical perspective, we can engineer the system perfectly, but at the end of the day, humans interact with these systems, and it's our responsibility to minimize bias and prejudice as much as possible. Algorithms are only as good as the data we provide them. Understanding the data and the context in which it's being used is the first step in battling bias, and this understanding will help you build better solutions because you'll be well versed in the problem space. Providing balanced data with as little bias as possible should result in better solutions and benefit more people.

The law, privacy, and consent: Knowing the importance of core values

The legal aspect of what we do is hugely important. The law governs what we can and can't do in the interest of society as a whole. Because many laws were written when computers and the internet were less important in our lives than they are today, we find many gray areas in how we develop technology and what we're allowed to do with that technology. That said, laws are slowly changing to adapt to rapid innovation.

We compromise our privacy almost every hour of every day via our interactions on computers, mobile phones, smart devices, and sensors, for example. We transmit a vast amount of information about ourselves, some more personal than others. How is that data being processed and stored? We should consider these facts when building solutions. People should have a choice about what personal data is captured, processed, and stored; how that data is used; and who can potentially access that data. In my experience, people generally accept solutions that use their data to improve the products they use and add value to their lives. Most important, people are more accepting when they're given a choice and that choice is respected.

Singularity: Exploring the unknown

The *singularity* is the idea that we create an AI system so generally intelligent that it's capable of improving itself and expanding its intelligence to a stage where it becomes super intelligence. The concern is that humans can't understand or control something of this magnitude, which could change civilization as we know it for reasons we can't even comprehend. Some people are concerned that this intelligence may see humans as a threat; others propose that we may be to a super intelligence what ants are to us. We don't pay explicit attention to ants or concern ourselves about how they live, but if we're irritated by them, we deal with them swiftly. Whether or not these assumptions are accurate representations of the future, we must be responsible and think about the decisions we make because they ultimately affect a real person, groups of people, or the world at large.

acknowledgments

Writing this book has been one of the most challenging yet rewarding experiences I've had. I had to make time when I had none, find the right headspace while juggling many responsibilities, and create motivation while being caught up in the reality of life. I've learned and grown tremendously through this experience. Thank you, Bert Bates, for being a fantastic editor and mentor during the writing of the first edition of this book. I learned so much about effective teaching and written communication from you. Our discussions and debates and your empathy throughout the process helped mold this book into what it is.

Every project needs someone who's organized, with a finger on the pulse, making sure that things are happening. For this, I'd like to thank Elesha Hyde, my development editor. Working with you has been an absolute pleasure. You always provide direction and interesting insights about my work.

We always need people to bounce ideas off, and who better to annoy than your friends? I'd especially like to thank Hennie Brink for always being a great sounding board and pillar of support.

For ensuring technical rigor and objective feedback, I'd like to thank technical editor Emmanuel Maggiori. Emmanuel is a 10-year AI industry insider and author of the book *Smart Until It's Dumb*. He has developed AI for a wide variety of applications, from extracting objects from satellite images to packaging holiday deals for millions of travelers every day.

I'd also like to thank review editor Dunja Nikitović, copy editor Keir Simpson, proofreader Jason Everett, typesetter Tamara Švelić Sabljić, and production editor Kathy Rossland.

Finally, I'd like to thank all the reviewers who took the time to read the manuscript of the second edition throughout development and provided invaluable feedback that made the book better in many ways: David Jacobs, Denise Sison, Evgueni Serdiouk, Frances Buontempo, Gavin Baumanis, James Dietrich, Johannes Ball, Joseph Catanzarite, Ken Byrne, Kévin Etienne, Kostas Passadis, Rodolfo Allendes, Sandra Pintor, Senzeyu Zhang, and Serhii Savin.

about this book

Grokking AI Algorithms was written and illustrated to make understanding and implementing AI algorithms and their uses in solving problems more accessible to the average person in the technology industry through the use of relatable analogies, practical examples, and visual explanations.

Who should read this book

Grokking AI Algorithms is for software developers and anyone in the software industry who wants to grasp the intuitions and uncover the workings and algorithms behind AI through practical examples and visual explanations instead of going through theoretical deep dives and mathematical proofs.

This book is aimed at anyone who understands basic computer programming concepts, including variables, data types, arrays, conditional statements, iterators, classes, and functions. Experience in any language is sufficient. The book is also for anyone who understands basic mathematical concepts such as data variables, the representation of functions, and how to plot data and functions on a graph.

A fundamental understanding of AI algorithms is becoming more useful for many roles in the software industry. Gone are the days when AI was solely the domain of research scientists and academic labs. Today, AI helps power searches, optimize delivery routes, and personalize user interfaces. As these tools become standard components of the software stack, engineers, product managers, and architects alike must look beyond the black box to understand the mechanics that drive them. By grasping the intuition behind these algorithms, we move from simply consuming APIs to building smarter, more resilient systems.

How this book is organized: A road map

This book contains 12 chapters, each focusing on a different AI algorithm or algorithmic approach. The early chapters cover fundamental algorithms and concepts that form a foundation for learning about more sophisticated algorithms later in the book.

- *Chapter 1, Intuition of AI*—Introduces the intuition and fundamental concepts that surround data, types of problems, categories of algorithms and paradigms, and use cases for AI algorithms
- *Chapter 2, Search fundamentals*—Covers the core concepts of data structures and approaches for primitive search algorithms and their uses
- *Chapter 3, Intelligent search*—Goes beyond primitive search algorithms and introduces search algorithms for finding solutions more optimally, as well as finding solutions in a competitive environment
- *Chapter 4, Evolutionary algorithms*—Dives into the workings of genetic algorithms in which solutions to problems are iteratively generated and improved upon by mimicking evolution in nature
- *Chapter 5, Advanced evolutionary approaches*—Continues the topic of genetic algorithms but tackles advanced concepts involving how steps in the algorithm can be adjusted to solve different types of problems more optimally
- *Chapter 6, Swarm intelligence: Ants*—Digs into the intuition for swarm intelligence and works through how the Ant Colony Optimization algorithm uses a theory of how ants live and work to solve hard problems
- *Chapter 7, Swarm intelligence: Particles*—Continues the topic of swarm algorithms while diving into what optimization problems are and how they're solved using Particle Swarm Optimization, seeking good solutions in large search spaces
- *Chapter 8, Machine learning*—Works through a machine learning workflow for data preparation, processing, modeling, and testing to solve regression problems with linear regression, and classification problems with decision trees
- *Chapter 9, Artificial neural networks*—Uncovers the intuition, logical steps, and mathematical calculations in training and using an artificial neural network (ANN) to find patterns in data and make predictions while highlighting its place in a machine learning workflow
- *Chapter 10, Reinforcement learning*—Covers the intuition of reinforcement learning from behavioral psychology and works through the Q-learning algorithm for agents to learn good and bad decisions to make in an environment
- *Chapter 11, Large language models*—Explores the data pipeline for training language models using the Transformer architecture, where text data is encoded in machine-

understandable numbers and meaning is found from relationships to generate intelligent text output
- *Chapter 12, Generative image models*—Covers the intuition and process for image generation using diffusion, showing how a model learns from a text prompt to mold random noise into a desired image

The chapters should be read from start to end sequentially. Concepts and understandings are built up as the chapters progress. It's useful to reference the Python code in the repository after reading each chapter to experiment with and gain practical insight into how the respective algorithm can be implemented.

About the code

This book contains Python code to focus on the intuition and logical thinking behind the algorithms. This code is to be treated as samples. For an in-depth look at working runnable Python code for all algorithms described in the book, visit the GitHub page https://github.com/rishal-hurbans/Grokking-Artificial-Intelligence-Algorithms. Setup instructions and comments are provided in the source code to guide you as you learn. One potential learning approach is to read each chapter and then reference the code to cement your understanding of the algorithms.

The Python source code is intended to be a reference for implementing the algorithms. These examples are optimized for learning and *are not for production use*. The code was written to serve as a teaching tool. Using established libraries and frameworks is recommended for projects that will make their way to production because they're usually optimized for performance, well tested, and well supported.

This book contains many examples of source code both in listings and inline with normal text. In both cases, source code is formatted in a `fixed-width font like this` to separate it from ordinary text. Sometimes, code is also **`in bold`** to highlight code that has changed from previous steps in the chapter, such as when a new feature adds to an existing line of code.

In many cases, the original source code has been reformatted; we've added line breaks and reworked indentation to accommodate the available page space in the book. In rare cases, even this was not enough, and listings include line-continuation markers (➥). Also, comments in the source code are often removed when the code is described in the text. Code annotations accompany many of the code samples, highlighting important concepts.

You can get executable snippets of code from the liveBook (online) version of this book at https://livebook.manning.com/book/grokking-ai-algorithms-second-edition. The complete code for the examples in the book is available for download on the Manning website at https://www.manning.com/books/grokking-ai-algorithms-second-edition and on GitHub at https://github.com/rishal-hurbans/Grokking-Artificial-Intelligence-Algorithms.

liveBook discussion forum

Purchase of *Grokking AI Algorithms* includes free access to liveBook, Manning's online reading platform. Using liveBook's exclusive discussion features, you can attach comments to the book globally or to specific sections or paragraphs. It's a snap to make notes for yourself, ask and answer technical questions, and receive help from the author and other users. To access the forum, go to https://livebook.manning.com/book/grokking-ai-algorithms-second-edition/discussion.

Manning's commitment to our readers is to provide a venue where meaningful dialogue between individual readers and between readers and the author can take place. It is not a commitment to any specific amount of participation on the part of the author, whose contribution to the forum remains voluntary (and unpaid). We suggest that you try asking the author some challenging questions lest his interest stray! The forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

Other online resources

- Source code for *Grokking AI Algorithms*, Second Edition: https://github.com/rishal-hurbans/Grokking-Artificial-Intelligence-Algorithms
- Author website: https://rhurbans.com

about the author

Rishal Hurbans is a technologist, entrepreneur, and author who is passionate about crafting innovative products and meaningful experiences. He holds a Bachelor of Science with Honors in Computer Science degree and has more than a decade of expertise delivering diverse technology solutions across the finance, health, agriculture, mining, and aviation industries.

Rishal is driven by a deep interest in making complex concepts accessible through visual, intuitive, and practical experiences. He has delivered dozens of keynotes and workshops around the world on AI, software engineering, and leadership. Rishal continues to build at the intersection of design, technology, and creativity, guided by the belief that technology should amplify human potential. For more about Rishal Hurbans, see https://rhurbans.com.

Intuition of AI 1

In this chapter

- Defining AI as we know it
- Gaining an intuition of concepts underpinning AI and important terminology
- Defining problem types and approaches to solving them
- Outlining the AI algorithms discussed in this book
- Exploring real-world use cases for AI algorithms

Artificial Intelligence, or AI, has become a foundational tool in modern software engineering. Developers use AI tools to create software, and many software applications now take advantage of content generated by large language models (LLMs), agents, and other AI-powered features. A new generation of easy-to-use tools, frameworks, and APIs offering quick access to sophisticated models makes it easier than ever to use AI without really understanding how it works. As a technology professional, though, it pays to have an idea what's going on inside the AI black box.

In this book, we'll discuss how different types of AI work as we explore the basic structures of their implementations. As we go, we'll trace the evolution of algorithmic intelligence, moving from the explicit logic of search and evolutionary algorithms through the statistical principles of machine learning and finally into the complex architectures of deep learning and generative AI. I hope that as you start to "grok" AI, you'll be better equipped to reason about, implement, and innovate with the systems that are reshaping the world. Now let's crack open the black box!

What is artificial intelligence?

Intelligence is a bit of a mystery. Philosophers, psychologists, and engineers view intelligence differently. Yet we recognize it everywhere: in the collective work of ants, the flocking of birds, and our own thinking and behavior. Great minds have long debated the nature of intelligence. Salvador Dalí saw it as *ambition*; Einstein tied it to *imagination*; Stephen Hawking defined it as the *ability to adapt*. Although there is no single agreed-on definition, we generally use human behavior as the benchmark.

At a minimum, intelligence means being *autonomous* and *adaptive*. Autonomous entities act without constant instruction, and adaptive ones adjust to changing environments. Whether the intelligence is biological or mechanical, its fuel is always data. The sights we see, the sounds we hear, and the measurements of our world are all inputs. We consume, process, and act on this data; therefore, understanding AI begins with understanding the data that powers it.

Defining AI

Generative AI, like natural language models and chatbots, has become the public face of AI. In research and practice, however, AI is far more diverse. The complex systems we use today are built on many foundational algorithms geared to solving different problems—from specific tasks such as searching information effectively to general challenges such as understanding language. Throughout this book, we'll explore these algorithms and their use cases and develop an *intuition*—a solid understanding and mental model—for modern machine learning, artificial neural networks (ANNs), and generative models.

For the sake of our sanity, let's define *AI* as systems that perform tasks typically requiring human intelligence. These tasks include simulating senses like vision and hearing, as well as mastering language to reason about complex problems. Here are some examples of tasks that exhibit AI:

- Playing and winning complex games
- Detecting cancer tumors from body-image scans

- Generating artwork based on a natural language prompt
- Driving cars autonomously
- Using chatbots encoded with the history of information on the internet

Douglas Hofstadter famously quipped, "AI is whatever hasn't been done yet." A calculator was once considered intelligent; now it's taken for granted. Whether an algorithm fits a strict academic definition matters less than its utility: if it solves a complex problem autonomously, it belongs in our toolkit.

The algorithms in this book have been classified as AI algorithms in the past or present. It doesn't matter whether they fit a specific definition of AI; what matters is that they're useful. Intuition about them forms a robust understanding from foundational to sophisticated applications.

Data is the fuel for AI algorithms

Data is the fuel that makes AI algorithms work. With the incorrect choice of data, badly represented data, or missing data, algorithms perform poorly, so the outcome is only as good as the data provided. The world is filled with data, and that data exists in forms that we can't even sense. Data can represent values that are measured numerically, such as the current temperature in the Arctic, the number of cats in a field, or a person's current age in days. All these examples involve capturing accurate numeric values based on facts. It's difficult to misinterpret this data: the temperature at a specific location at a specific point in time, for example, is absolutely true and not subject to any bias. This type of data is known as *quantitative data*. But cherry-picking (or sampling) specific data points intentionally or unintentionally can create bias.

Data can also represent values of observations, such as the smell of a flower or one's subjective review of a movie. This type of data is known as *qualitative data*. Sometimes, this data is difficult to interpret because it's not an absolute truth but a perception of someone's truth. Figure 1.1 illustrates some of the quantitative and qualitative data around us.

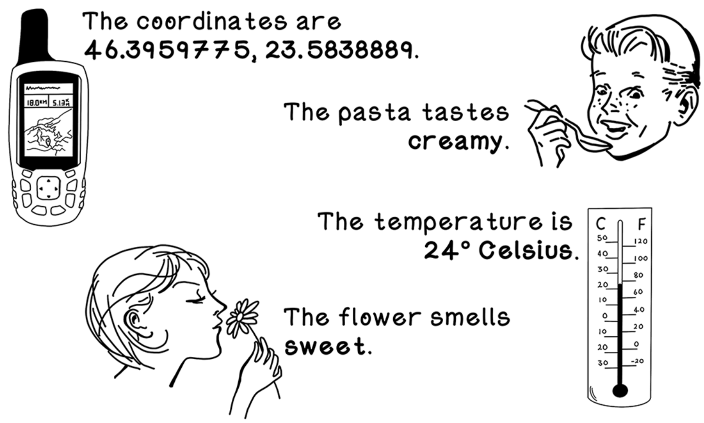


Figure 1.1 Examples of data around us

Data is raw facts about things, so recordings of it should have no bias. In the real world, however, data is collected, recorded, and related by people based on a specific context with a specific understanding of how the data may be used. Constructing meaningful insights to answer questions based on data creates *information*. Furthermore, consciously applying information in conjunction with experiences creates *knowledge*, which is partly what we try to simulate with AI algorithms. Consider, for example, how the following concepts apply to a hospital patient:

- *Data*—The patient's temperature is 38°C.
- *Information*—The patient has a fever.
- *Knowledge*—We should administer medication to lower the fever.

Figure 1.2 shows how quantitative and qualitative data can be interpreted. Standardized instruments such as clocks, calculators, and scales are usually used to measure quantitative data, whereas our senses of smell, sound, taste, touch, and sight, as well as our subjective thoughts, are usually used to create qualitative data.

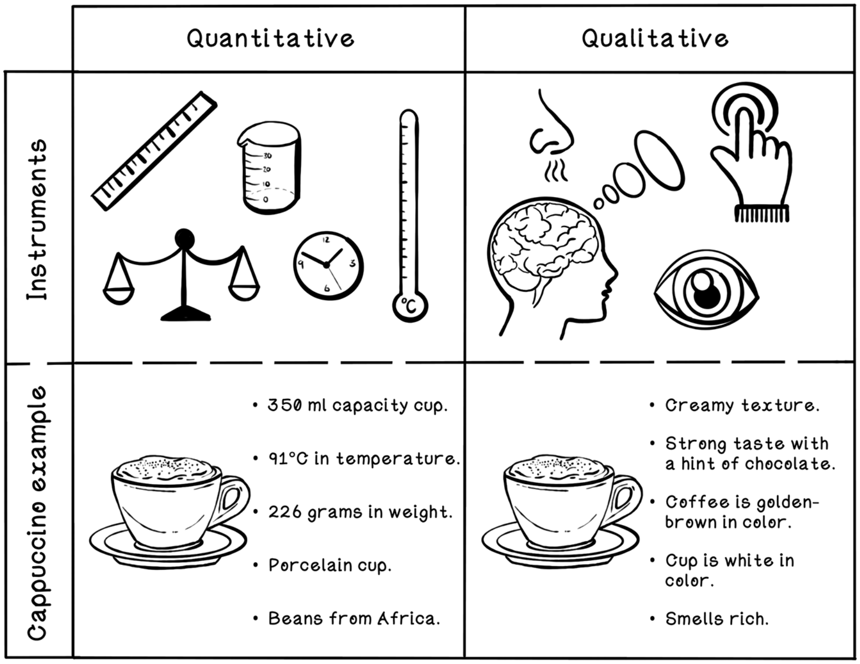


Figure 1.2 Qualitative data vs. quantitative data

Data, information, and knowledge can be interpreted differently based on a person's level of understanding of that domain and their outlook on the world. This fact has consequences for the quality of solutions that we build, making the "scientific method" aspect of creating technology hugely important. By following repeatable scientific processes to capture data, conduct experiments, and report findings accurately, we can strive to achieve more accurate results and better solutions when processing data with algorithms.

Algorithms are like recipes

Now we have a loose definition of AI and an understanding of the importance of data. Because we'll explore several AI algorithms throughout this book, it's useful to understand exactly what an algorithm is. An *algorithm* is a set of instructions and rules provided as a specification to accomplish a goal. Algorithms typically accept inputs, and after several finite steps in which the algorithm progresses through varying states, an output is produced.

Even a task as simple as reading a book can be represented as an algorithm. Here's an example of the steps involved in reading this book:

1. Find the book *Grokking AI Algorithms*.
2. Open the book.
3. While unread pages remain,
   a. Read a page.
   b. Think about what you've learned.
   c. Turn to the next page.
4. Think about how you can apply your learning in the real world.

An algorithm can be compared to a recipe (figure 1.3). Given ingredients and tools as inputs and instructions for creating a specific dish, a meal is produced as the output.

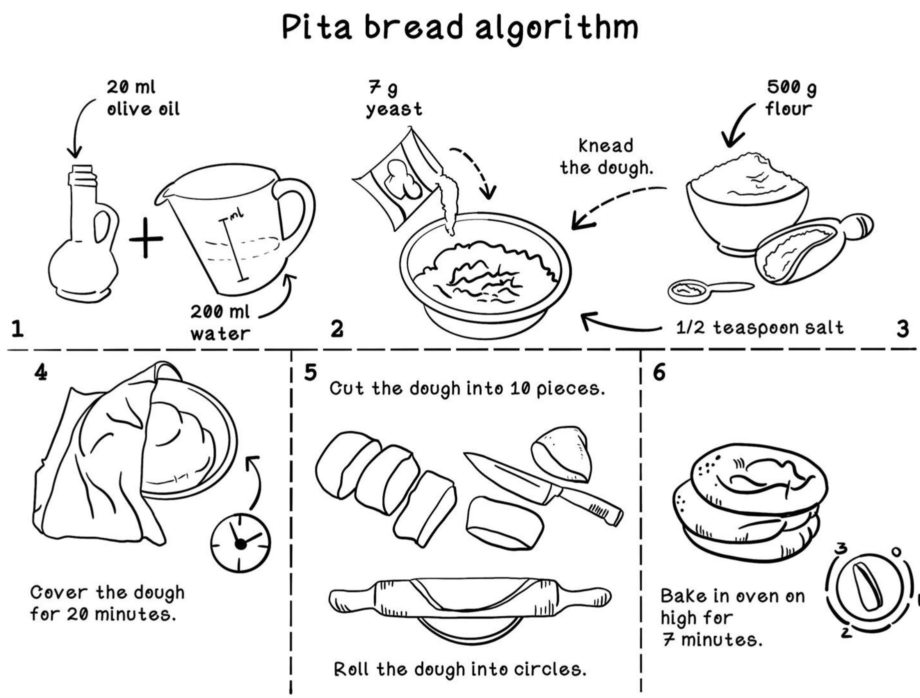


Figure 1.3 An example showing that an algorithm is like a recipe

Algorithms are used in many solutions. We can enable live video chat across the world through compression algorithms, for example, and we can navigate cities through map applications that use real-time routing algorithms. Even a simple "Hello World" program has many algorithms at play that translate the human-readable programming language into machine code and execute the instructions on specific hardware. You can find algorithms everywhere if you look closely enough.

To illustrate something more closely related to the algorithms in this book, figure 1.4 shows a number-guessing-game algorithm represented as a flow chart. The computer generates a random number in a given range, and the player attempts to guess that number. The algorithm has discrete steps that perform an action or determine a decision before moving to the next operation. You'll see flow charts like the one in figure 1.4 throughout the book to explain the life cycles of algorithms.

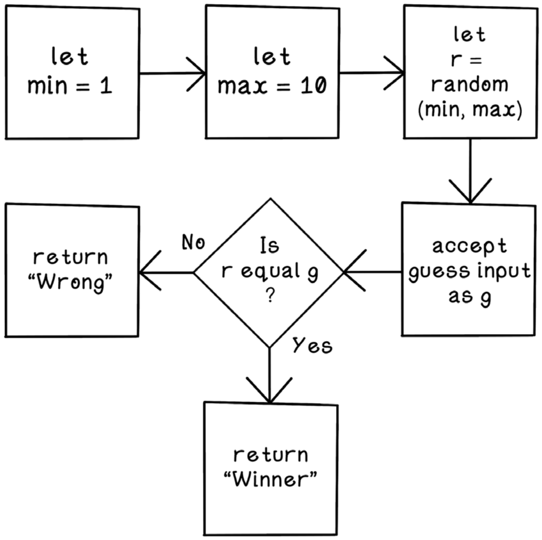


Figure 1.4 A number-guessing-game algorithm flow chart

Algorithms vs. models

Algorithms are the logic; models are the result. To understand how AI systems are built, we must distinguish between the process (the recipe) and the representation (the meal). In this book, we'll see two distinct patterns depending on the type of AI we're using.

The algorithm as the active solver

Some algorithms actively solve a problem in real-time. A search algorithm, for example, is the star of the show for navigating a map or calculating the best move in a game. Search algorithms are deployed to production and run fresh every time to solve problems based on the current situation.

The algorithm as the builder

In machine learning and deep learning, the relationship changes. Here, the algorithm is a builder; its job is to look at data and construct a representation of the world. The algorithm is used to train a model, and the model is the artifact that is eventually deployed to production. This model contains the intelligence—pretrained by the algorithm—that is used for problem-solving.

The evolution of AI

A look back at the strides made in AI is useful for understanding that old techniques and new ideas can be harnessed to solve problems in innovative ways.

AI isn't a new idea. History is filled with myths of mechanical men and autonomous "thinking" machines. Looking back, we find that we're standing on the shoulders of giants. Perhaps we ourselves can contribute to the pool of knowledge in a small way.

Looking at past developments highlights the importance of understanding the fundamentals of AI; algorithms from decades ago are critical in many modern AI implementations. This book starts with fundamental algorithms that help build the intuition of problem-solving and gradually moves to more complex and modern approaches.

Figure 1.5 isn't an exhaustive list of achievements in AI—simply a small set of examples. History is filled with many more breakthroughs.

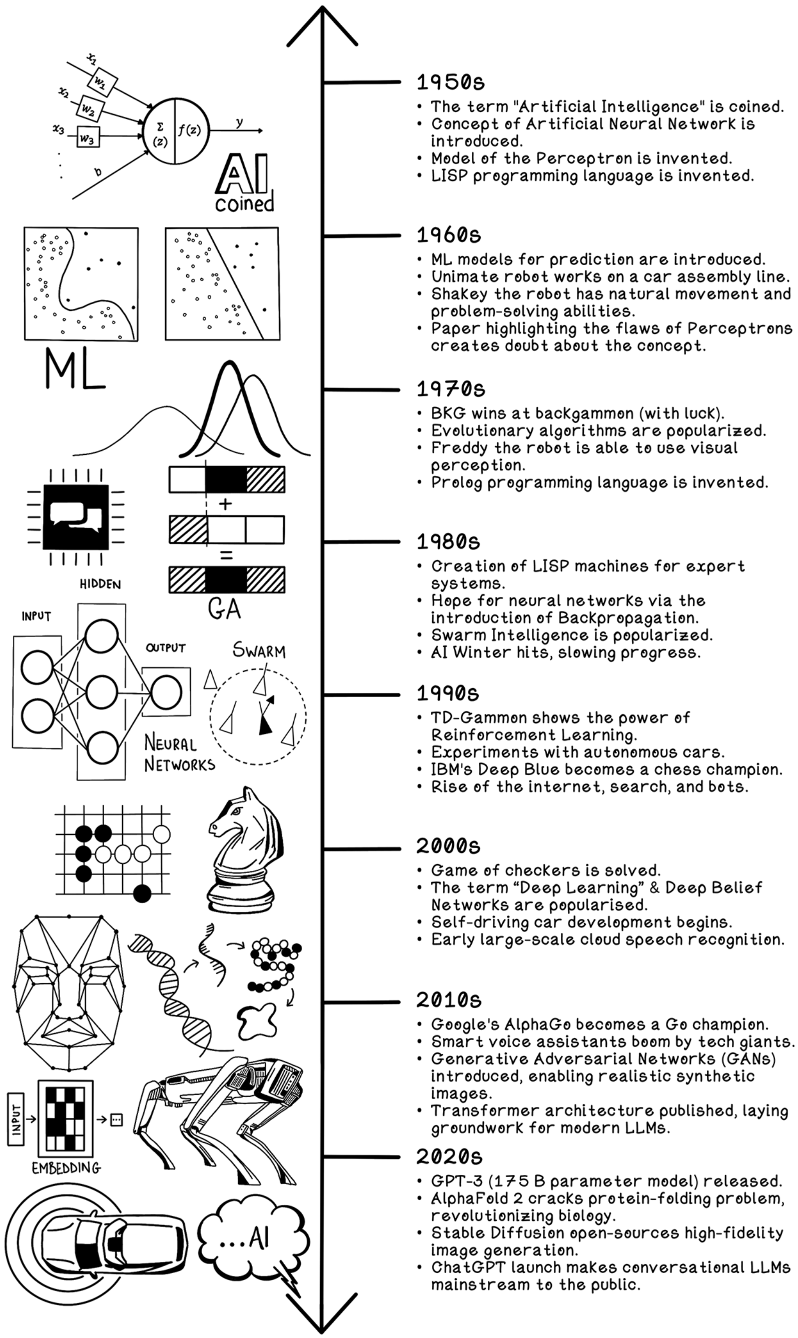


Figure 1.5 The evolution of AI

Different types of problems

AI algorithms are powerful, but they're not silver bullets that can solve any problem (yet). But what are these "problems"? This section looks at types of problems that we commonly experience in computer science. This intuition can help us identify these problems in the real world and guide the choice of algorithms we use. A grasp of this terminology will be useful as we progress through the book.

Search problems: Finding a path to a solution

A *search problem* involves a situation that has multiple possible solutions, each of which represents a sequence of steps (path) toward a goal. Some solutions contain overlapping paths to success; some are better than others; some are cheaper to achieve than others. A "better" solution is determined by the specific context; a "cheaper" solution means computationally cheaper to execute.

Consider the example of determining the shortest path between cities on a map. Many routes may be available, with different distances and traffic conditions, but some routes are better than others. Many AI algorithms are based on the concept of searching a space of possible solutions.

Optimization problems: Finding a good solution

An *optimization problem* involves a situation in which a vast number of valid solutions exists and the best solution is difficult to find. This type of problem usually has an enormous number of possible solutions, which differ in how well they solve the problem.

An example is packing luggage in the trunk of a car. The goal is to maximize the use of space while adhering to strict constraints, such as the fixed dimensions of the trunk or a weight limit. Many combinations are available, but valid solutions must fit within these boundaries. If the trunk is packed effectively, more luggage can fit in it.

Local best vs. global best

Because optimization problems have many solutions and because these solutions exist at different points in the search space, the concepts of local bests and global bests come into play. A *local best* solution is the best solution within a specific area in the search space, and a *global best* is the best solution in the entire search space. The challenge for AI is avoiding the trap of a local best (a good solution) while missing the global best (the perfect solution).

Suppose that you're searching for the best restaurant. You may find the best restaurant in your local area, but it may not necessarily be the best restaurant in the country or the best restaurant in the world. The best in the world is the global best.

Prediction and classification problems: Learning from patterns in data

Prediction problems occur when we have data about something and want to try to find patterns and predict a numerical value. We might have data about different vehicles and their engine sizes, as well as each vehicle's fuel consumption. Can we predict the fuel consumption of a new model of vehicle, given its engine size? If the historical data shows a correlation between engine sizes and fuel consumption, this prediction is possible.

Classification problems are similar to prediction problems, but instead of trying to find an exact value such as fuel consumption, we try to find a category. If we have the physical dimensions of a vehicle, its engine size, and the number of seats, can we predict whether that vehicle is a motorcycle, sedan, or sport-utility vehicle? Classification problems require finding patterns in the data that group examples into categories.

> **TIP** Here's a helpful rule of thumb for remembering the difference: prediction outputs a quantity (a number), whereas classification outputs a label (a category).

Clustering problems: Identifying patterns in data

Clustering problems are scenarios in which we uncover trends and relationships from data, using different aspects of the data to group examples in different ways. Given cost and location data about restaurants, for example, we may find that younger people tend to frequent locations where the food is cheaper. Clustering aims to find relationships in data even when a precise question isn't being asked. This approach is also useful for gaining a better understanding of data to inform what we might be able to do with it.

Deterministic models: Getting the same result each time

Deterministic models are models that, given a specific input, return a consistent output. If you input 100°C into a unit conversion model, the output will always be 212°F. It doesn't matter what time of day it is, where you are, or how many times you repeat the calculation; the relationship is fixed, and the result never varies.

Probabilistic models: Getting potentially different results each time

Probabilistic models are models that, given a specific input, return an outcome from a set of possible outcomes. Probabilistic models usually have an element of controlled randomness that contributes to the possible set of outcomes. If you type the phrase "The best pet is a…" on your phone, the device might autocomplete various words based on words you've used in the past. It might assign *cat* 40%, *dog* 35%, and *goldfish* 25%, for example. If you run this model once, it might pick *cat*. If you run the model again, the element of randomness might cause it to pick *dog*. The input is the same, but the output varies based on the probability distribution.

Intuition of AI concepts

Trying to make sense of different but similar words and concepts in AI can be daunting. In this section, we demystify them and form a road map of the topics covered in this book. Let's dive into the levels of AI, introduced in figure 1.6.

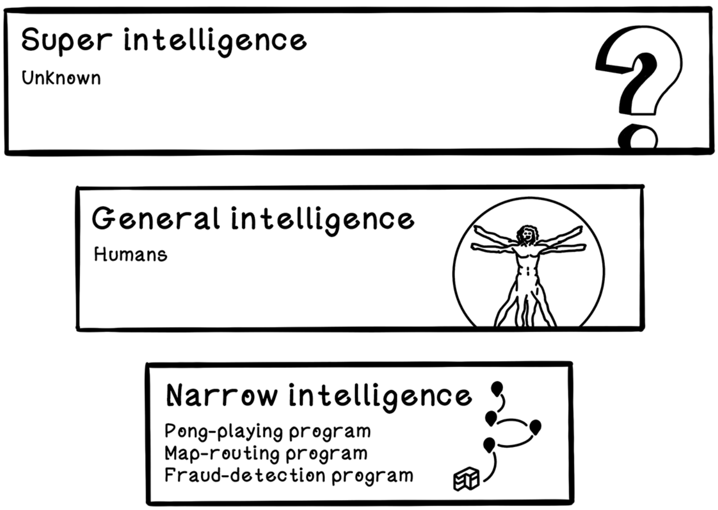


Figure 1.6 Levels of AI

Narrow intelligence: Specific-purpose solutions

Artificial narrow intelligence (ANI) systems specialize in a single domain and can't transfer knowledge. A model trained to analyze spending behavior, for example, can't identify cats in an image. But distinct narrow systems can be combined to simulate broader intelligence. A voice assistant, for example, stacks speech recognition, web search, and recommendation algorithms to interact naturally with users. Different narrow intelligence systems can be combined in sensible ways to create something greater. A modern voice assistant isn't one smart brain; it's a combination of distinct narrow models (speech-to-text + web search + audio generation) working together to simulate a conversation.

General intelligence: Humanlike solutions

Artificial general intelligence (AGI) mirrors human adaptability: the ability to apply knowledge from one problem to another. If you felt pain when touching something extremely hot as a child, you can extrapolate: you know that other hot things have a chance of hurting you. Just as a child generalizes that "extremely hot" implies "danger" across different objects, AGI integrates memory, reasoning, and sensory input to solve novel problems. Unlike the voice assistant, which would require a code update to learn a new skill, an AGI could simply figure it out. If you asked a narrow assistant to perform a task it wasn't programmed for (such as negotiating a better price for your flight), it would fail. An AGI, however, would understand the intent of "saving money," apply reasoning from other contexts, and attempt the negotiation. Although true AGI remains elusive, modern LLMs represent a significant step toward this flexible reasoning.

Super intelligence: The great unknown

Artificial super intelligence (ASI) refers to systems that surpass the brightest human minds in every field. Although ASI is often depicted in science fiction as apocalyptic, the core definition is simply intelligence beyond our comprehension. Whether humans can create something smarter than ourselves remains a philosophical debate, making ASI a realm of pure speculation for now.

Old AI and New AI

Sometimes, we use the notions of Old AI and New AI.

Old AI is explicit logic and search that relies on humans to encode the rules of the world. The machine doesn't learn; it calculates solutions based on logic and rules provided by programmers. A classic example is the Minimax algorithm in chess: the human defines how pieces move and how to score a board position, and the AI uses computational power and smart branching to search future moves and find the best one.

New AI learns from data and flips the Old AI approach. Instead of being told what the rules are or how to score a situation, these models analyze vast datasets to figure things out for themselves. A modern neural network playing chess, for example, isn't following hardcoded heuristics; it has played millions of games against itself to learn patterns of victory that human programmers might not even understand.

We learn about both categories because, although search algorithms are often categorized as Old AI, the algorithms aren't obsolete; in fact, they're often paired with modern techniques to solve difficult problems. LLMs use search strategies to determine the best sequence of words to generate, for example. You need to understand the logic of search before you can understand how a neural network searches for an optimal solution. Figure 1.7 illustrates the relationships among several concepts within AI.

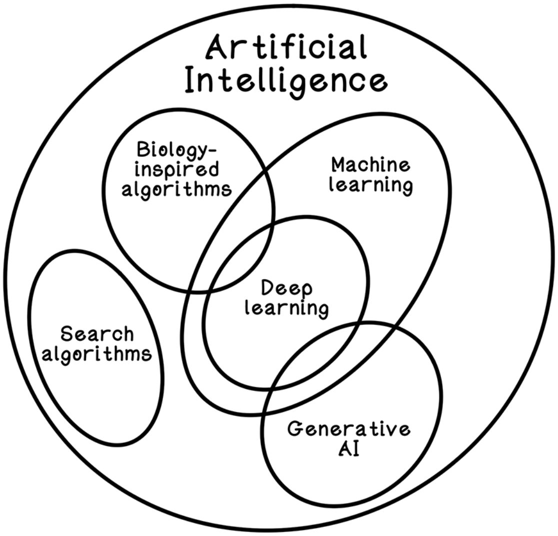


Figure 1.7 Categorization of concepts within AI

Search algorithms

Search algorithms are the bedrock of problem-solving. They're essential when a goal requires taking a sequence of actions, such as navigating a maze or calculating the winning move in chess. Instead of brute-forcing every possibility, which could take thousands of hours, smart search algorithms evaluate future states to find the optimal path efficiently. We start our journey here. Chapter 2 covers the fundamental search algorithms, and chapter 3 dives into intelligent search methods that strategize rather than guess.

Biology-inspired algorithms

When we look at the world around us, we notice incredible things happening in creatures, plants, and other living organisms, such as the cooperation of ants in gathering food, the flocking of birds when migrating, the estimation of how organic brains work, and the evolution of organisms to produce stronger offspring. By observing and learning from these phenomena, we've gained knowledge of how these organic systems operate and how simple rules can result in incredible emergent intelligent behavior.

Consider two types of biology-inspired algorithms: evolutionary and swarm. Evolutionary algorithms, inspired by Darwinian theory, use reproduction and mutation to evolve code that improves over generations. We explore this survival of the fittest in chapters 4 and 5.

Swarm intelligence mimics the collective power of "dumb" individuals acting smart as a group. Chapter 6 explores intelligent pathfinding inspired by ants, and chapter 7 covers solving optimization problems inspired by the way animals flock.

Machine learning algorithms

Chapter 8 takes a statistical approach to training models to learn from data. The umbrella of machine learning has a variety of algorithms that we can harness to improve our understanding of relationships in data, make decisions, and make predictions based on that data. There are three main approaches in machine learning:

- *Supervised learning* means training models with algorithms when the training data has known outcomes for a question being asked, such as determining the type of fruit if we have a set of data that includes the weight, color, texture, and fruit label for each example.
- *Unsupervised learning* uncovers hidden relationships and structures within the data that guide us to ask the dataset relevant questions. It may find patterns in properties of similar fruits and group them accordingly, which can inform the exact questions we want to ask the data. These core concepts and algorithms help us create a foundation for exploring advanced algorithms in the future.
- *Reinforcement learning* is inspired by behavioral psychology. In short, it describes rewarding a person if they perform a useful action and penalizing them if they perform an unfavorable action. When a child achieves good results on their report card, they're usually rewarded, but poor performance sometimes results in punishment, reinforcing the behavior of achieving good results. Chapter 10 explores how reinforcement learning is used to train intelligent models.

Deep learning algorithms

Deep learning is a broader family of approaches and algorithms that are used to achieve narrow intelligence and strive for general intelligence. Deep learning usually implies that the approach is attempting to solve a problem in a more general way, such as linguistic intelligence or spatial reasoning, or is being applied to problems that require more generalization, such as speech recognition or computer vision. Deep learning approaches usually employ many layers of ANNs. By using different layers of intelligent components, each layer solves specialized problems; together, the layers solve complex problems toward a greater goal. Identifying any object in an image, for example, is a general problem, but it can be broken into understanding color, recognizing shapes of objects, and identifying relationships among objects to achieve a goal. We'll dive into how artificial neural networks operate in chapter 9.

Generative models

Generative AI marks the shift from analyzing existing data to creating new content. Instead of just classifying an image or predicting a number, these models generate text, code, and visuals that never existed before.

Large language models

Large language models (LLMs) are designed to master context, conversation, and reasoning. Chapter 11 explores the architecture that revolutionized this field—the Transformer—and explains how these models achieve human-level fluency.

Generative image models

Generative image models learn to sculpt pure random noise into structured, high-fidelity artwork. Chapter 12 dives into the mechanics of creativity, looking at how diffusion models and U-Nets operate.

Some uses for AI algorithms

The uses for AI techniques are potentially endless. The applications of AI are limited only by the availability of data. Wherever a complex problem and historical data exist, potential for optimization exists. This section explores how AI transforms different industries.

Agriculture: Optimizing plant growth

- *Problem*—One of the most important industries that sustain human life is farming. Farming is a high-stakes balancing act. A single crop's success depends on hundreds of interacting variables, including soil pH, moisture levels, microbial health, and unpredictable weather patterns. It's impossible for a human to perfectly calculate how these factors interact in real time.

- *Solution*—Modern farms use sensors to capture the environment as data. AI algorithms analyze these vast datasets to find hidden patterns, identifying exactly which combination of water, fertilizer, and timing results in the highest yield. Instead of guessing, farmers get real-time, data-driven recommendations to maximize growth and minimize waste (figure 1.8).

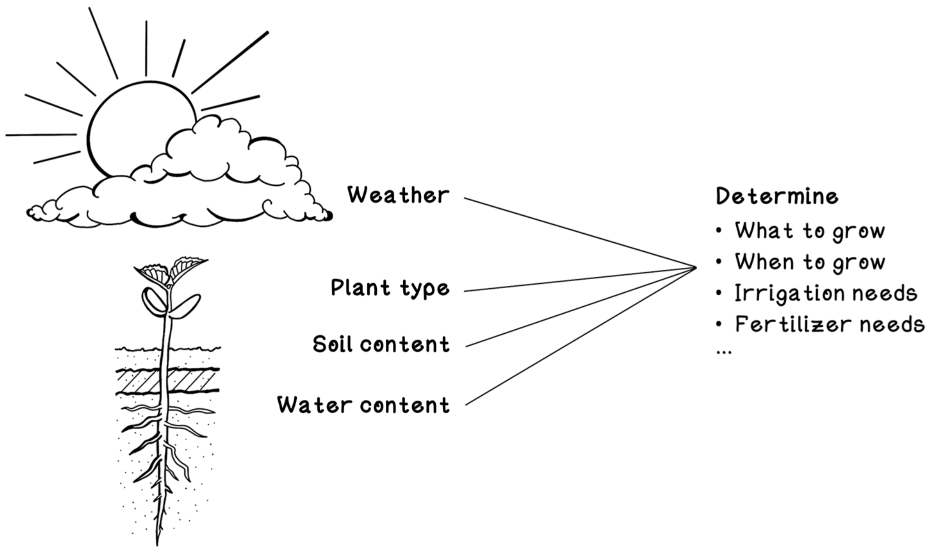


Figure 1.8 Using data to optimize crop farming

Banking: Preventing fraudulent transactions

- *Problem*—As banking moved from physical branches to digital networks, the volume of transactions exploded. With millions of payments happening every second, it's impossible for humans to review them manually for suspicious activity. Furthermore, static security rules such as "Flag all transactions over $10,000" are rigid and easy for clever criminals to bypass by moving $9,999 instead.
- *Solution*—This example is a classic case of anomaly detection. AI algorithms analyze a user's historical data to learn their specific "pattern of life"—where they shop, how much they spend, and at what times. When a transaction deviates from this learned baseline, such as by making a sudden high-value purchase in a foreign country, the model flags it within milliseconds, stopping the potential theft before the money leaves the account.

Cybersecurity: Safeguarding email inboxes

- *Problem*—Email scams are primary entry points for cyberattacks. In the past, spam filters relied on blacklists of bad words (such as *lottery* and *winner*). But scammers easily bypassed blacklists by changing spellings or using social-engineering tactics that looked like legitimate business requests.
- *Solution*—Modern AI uses natural language processing (NLP) to read emails. Instead of looking for keywords like *lottery*, AI analyzes the context and intent of the message. It can determine that an email request for gift cards that seems to come from your CEO is suspicious, even if it contains no typos: it's suspicious based on the sender's writing style and the unusual request. This approach keeps inboxes clean and prevents users from clicking dangerous links.

Health care: Diagnosing patients

- *Problem*—Radiologists and doctors must review thousands of X-rays, MRIs, and CT scans to detect diseases. Fatigue can lead to human error, and subtle early signs of conditions such as cancer or pneumonia can be imperceptible to the naked human eye.
- *Solution*—This scenario is a prime application for computer vision. AI models are trained on millions of medical images to recognize the visual textures of disease. These systems can scan images in seconds, highlighting potential tumors or fractures with high accuracy (figure 1.9). They act as a second set of eyes, ensuring that doctors don't miss critical diagnoses and enabling treatment to begin earlier.

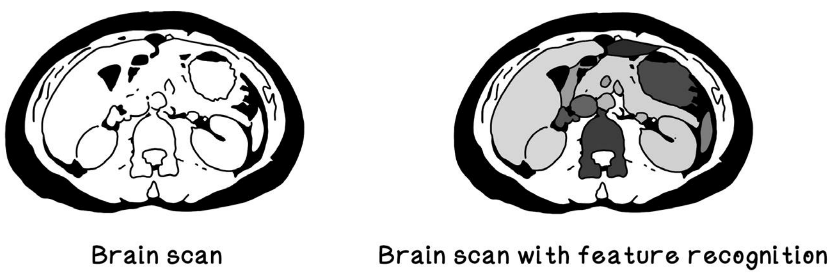


Figure 1.9 Using machine learning for feature recognition in brain scans