FROM SCRATCH

BUILD A Reasoning Model

Sebastian Raschka

MANNING

Build a Reasoning Model (From Scratch)

Sebastian Raschka

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
PO Box 761
Shelter Island, NY 11964

Development editor: Dustin Archibald
Technical editor: David Caswell
Review editor: Kishor Rit
Production editor: Kathy Rossland
Copy editor: Andy Carroll
Proofreader: Katie Tennant
Typesetter: Tamara Švelić Sabljić
Cover designer: Marija Tudor

ISBN 9781633434677
Printed in the United States of America

To Liza, my partner in every adventure; to my beloved family; and to the worldwide community of programmers and creators who have shaped my journey as an author

brief contents

contents

preface

More than a decade ago, my journey into AI began with statistical pattern classification, where I learned how far relatively simple models can go when they capture useful structure in data. Over time, that curiosity shifted from models that recognize patterns to models that can explain, plan, and solve multistep problems.

With the release of ChatGPT in 2022, large language models (LLMs) moved into the mainstream. A later shift, especially visible in late 2024 and throughout 2025, was the extension of LLMs to create so-called reasoning models, which can solve more complex problems, especially in technical domains such as math and code.

At the same time, reasoning models can feel opaque, and this book is my attempt to make reasoning models more approachable. Rather than training a giant model from scratch, we'll begin with a small pretrained LLM and apply reasoning techniques step by step. Along the way, you will implement text generation, evaluation with verifiers, inference-time scaling methods, and training methods, such as reinforcement learning with verifiable rewards and distillation from scratch. By the end, you will understand how the main reasoning methods work in practice and how they fit into a modern LLM development workflow.

Like my earlier book, *Build a Large Language Model (From Scratch)*, this one takes a code-first approach, but the focus here is different. Instead of explaining the Transformer architecture in depth or covering large-scale pretraining, here we'll concentrate on the methods that turn a conventional LLM into a reasoning model. If you have read the earlier book, this one should feel like a natural next step. If you haven't, you can still follow along here without needing all the underlying LLM details up front.

We'll use math examples throughout the book because they are practical for reasoning research and easy to verify automatically. That makes them a useful test bed for

understanding whether a model is actually solving a problem correctly. More broadly, my goal is not to teach one narrow benchmark but to give you the tools and intuition needed to build, evaluate, and improve reasoning systems in your own work.

When I first started working on reasoning models, I had to piece together ideas from papers, repositories, and scattered implementation notes. I hope this book saves you that effort and gives you a clearer path from first principles to working code. I strongly believe that the best way to understand reasoning models is to build one yourself.

Happy reading and coding!

acknowledgments

Writing a book is a significant undertaking, and I would like to express my sincere gratitude to my wife, Liza, for her patience and support throughout this process. Her unconditional love and constant encouragement have been absolutely essential.

I am incredibly grateful to Daniel Kleine and Dmitry Labazkin, whose invaluable feedback on the draft chapters and code went far above and beyond. Daniel's keen eye for detail and style, along with Dmitry's sharp insights into code performance, have undoubtedly made this book a smoother and more enjoyable read for all.

I would also like to thank the wonderful team at Manning Publications. Michael Stephens, thank you for the many productive conversations that helped shape the direction of this book. Dustin Archibald, I greatly appreciate your constructive guidance on adhering to Manning's guidelines, as well as your flexibility in accommodating the unique demands of this unconventional from-scratch approach. Special thanks also go to Aleksandar Dragosavljevic and Ivan Martinović for their work on layout and formatting. Finally, technical editor David Caswell, thank you for your thoughtful and generous feedback throughout the process. David is an industry-leading consultant focused on applying LLMs to journalism. He previously led AI innovation at the BBC, Tribune Publishing, and Yahoo!, he publishes peer-reviewed research on AI automation of information workflows, and he is a frequent speaker and writer on LLMs in the emerging AI-mediated information ecosystem.

Finally, I extend my thanks to the reviewers Alejandro Cuevas Rivero, Muhammad Ali Shafique, Arun Prakash A, Christopher Brousseau, David Curran, Fabio Montagna, Hamza Farooq, Hongming Zheng, Julien Thomazo, Lindo William Khoza, Michael Anthony Garcia, Naman Dwivedi, Pere Martra, Pradeep Dasigi, Salvatore Raieli, Sifal Klioui, Syed Baqir Ali, Thomas Viehmann, Tianrui Liu, Toni Ramchandani, and Viton

Vitanis for their thorough feedback on the drafts. Your keen eyes and insightful comments have been essential in improving the quality of this book.

To everyone who has contributed to this journey, I am sincerely grateful. Your support, expertise, and dedication have been instrumental in bringing this book to fruition. Thank you!

about this book

Build a Reasoning Model (From Scratch) is a hands-on guide to understanding how modern reasoning LLMs work by implementing their core techniques yourself. Rather than training a base model from the ground up, this book starts with a pretrained open source base LLM and shows you how to extend it step by step with reasoning capabilities. Along the way, you will learn how to generate text with a base model, evaluate answers with verifiers, improve performance through inference-time scaling, train with reinforcement learning, and distill reasoning behavior into smaller models. By the end of the book, you will understand how the main reasoning methods fit together in a practical LLM development workflow and how to build small but functional reasoning systems of your own.

The book takes a code-first approach. The "from scratch" part refers to implementing the reasoning methods yourself so that the underlying mechanics become transparent. If you are interested in how base LLMs are built and pretrained in detail, my earlier book, *Build a Large Language Model (From Scratch)* (http://mng.bz/M96o), provides that foundation, but it is not required for reading this book.

Who should read this book

Build a Reasoning Model (From Scratch) is for machine learning enthusiasts, engineers, researchers, students, and software developers who want a practical understanding of how reasoning models work and how to implement the main techniques behind them. It is written for readers who want to go beyond high-level descriptions and see how these systems are built, evaluated, and improved in code.

The most important prerequisite is solid Python experience. Prior exposure to machine learning, deep learning, or large language models is helpful, but you do not

need to be an expert. The book is intended to be approachable for motivated beginners while still offering enough technical depth for experienced practitioners.

The book uses math examples frequently because they are practical for reasoning research and easy to verify automatically. However, advanced mathematics is not required. Familiarity with basic algebra and a general comfort level with vectors and matrices is useful, though, to follow some of the code implementations.

Some prior PyTorch experience may help you move more quickly through the code, but it is not required. The focus of the book is on understanding reasoning methods and implementing them clearly, not on mastering every detail of a deep learning framework. Readers who have already read *Build a Large Language Model (From Scratch)* may recognize some broader LLM concepts, but this book is fully self-contained.

How this book is organized: A road map

This book is organized around a practical development pipeline for reasoning models. We'll begin with a conventional pretrained LLM, add evaluation so that we can measure progress, and then explore two main ways of improving reasoning: inference-time methods and additional training. Because later chapters build directly on the code and ideas introduced earlier, the book is best read in sequence.

Chapter 1 introduces the practical meaning of reasoning in the context of LLMs. It explains how reasoning models differ from conventional LLMs, it reviews the standard LLM training pipeline at a high level, and it introduces the main approaches used to improve reasoning.

Chapter 2 establishes the starting point by loading and using a pretrained base LLM. It covers the coding environment, tokenization, step-by-step text generation, and practical speedups such as key-value caching and model code compilation.

Chapter 3 adds evaluation. It shows how to extract final answers reliably, verify correctness against reference solutions, and build an evaluation pipeline for math-oriented reasoning tasks.

With that foundation in place, chapters 4 and 5 focus on inference-time scaling, improving reasoning without changing the model weights. Chapter 4 covers prompt-based reasoning, diverse decoding, and self-consistency. Chapter 5 extends this direction with self-refinement, simple scoring strategies, confidence estimation, and iterative answer improvement.

Chapters 6 through 8 implement training-time methods. Chapter 6 introduces reinforcement learning for reasoning models, with a focus on reinforcement learning with verifiable rewards (RLVR) and Group Relative Policy Optimization (GRPO). Chapter 7 builds on that implementation by covering practical monitoring, common training-failure modes, reward hacking, and useful extensions such as KL regularization and response-format rewards. Chapter 8 concludes the main text with distillation, showing how to create teacher-generated reasoning datasets, train a smaller student model, and evaluate the distilled result.