

Table of Contents

Preface	xix
Free Benefits with Your Book	xxv
Part 1: Understanding and Exploring DeepSeek	1
Chapter 1: What Is DeepSeek?	3
Introducing DeepSeek	4
Understanding the technical breakthroughs of DeepSeek	7
The training process • 8	
Reinforcement learning • 14	
Architecture modifications • 15	
<i>Comparison of major LLM architectures (2025)</i> • 17	
MoE architecture • 19	
<i>How DeepSeek implements MoE</i> • 19	
Training dataset and philosophy • 23	
Versions and evolution of DeepSeek	24
DeepSeek’s evolving ecosystem • 24	
Deep dive: Feature comparison of each model • 25	
DeepSeek product ecosystem • 28	
Platform integrations and deployment ecosystem • 29	
DeepSeek’s impact on the global AI ecosystem	30
Market disruption and price wars • 30	

Sparking the next wave of open source AI • 31	
A symbol of the technological maturity of China • 32	
Controversies surrounding DeepSeek • 33	
<i>Potential for bias and misuse</i> • 33	
<i>Openness and national security concerns</i> • 34	
<i>Cultural and political perceptions</i> • 35	
<i>Academic and industrial pushback</i> • 35	
<i>Ethical debates around RL methods</i> • 36	
<i>Discourse as a feature, not a flaw</i> • 36	
Summary	37
Chapter 2: Deep Dive into DeepSeek	39
Key architectural components of DeepSeek	39
Prompt routing in DeepSeek • 40	
Decoder • 42	
<i>Internal mechanics of the decoder module</i> • 44	
<i>Working example</i> • 50	
Mixture of experts • 52	
<i>How are experts initialized and configured?</i> • 52	
<i>How does gating compute which experts to activate?</i> • 54	
<i>How are experts activated?</i> • 54	
Multi-head latent attention: Memory-efficient attention for long contexts • 55	
DeepSeek features for efficient and context-aware responses • 57	
<i>FP8 training and precision control</i> • 57	
<i>Multi-token prediction for strategic decoding</i> • 58	
<i>GRPO and advanced RLHF</i> • 59	
<i>Chain-of-thought reasoning: CoT and long CoT</i> • 61	
<i>Choice of datasets</i> • 61	
<i>Test-time scaling</i> • 62	
Understanding the reasoning mechanics of DeepSeek	62
DeepSeek's thinking • 64	

<i>The think and answer blocks</i> • 64	
<i>Evaluating response quality with GRPO</i> • 65	
<i>Rule-based RLHF</i> • 67	
<i>How DeepSeek handles complex scenarios</i> • 68	
DeepSeek training • 69	
<i>Cold start data for training</i> • 70	
<i>The training pipeline of R1</i> • 73	
Distillation and distribution • 77	
<i>Emergent patterns in self-teaching</i> • 78	
Advanced capabilities of DeepSeek 78	
Vision capabilities of DeepSeek • 79	
<i>Architectural changes for vision-language integration</i> • 79	
Agentic reasoning and tool integration • 81	
DeepSeek in the global LLM landscape 82	
DeepSeek’s limitations and how they compare to other models • 85	
Scaling challenges and sparse expertise limitations • 85	
Inference latency and real-time interaction trade-offs • 86	
Interpretability and alignment risks • 86	
Context length and compression ceiling • 86	
Dataset and cultural scope gaps • 87	
Outlook on limitations • 87	
Summary 87	
Chapter 3: Prompting DeepSeek 89	
Technical requirements 91	
Core mental models and principles of DeepSeek 91	
Why structured, minimal prompts “click” with the R series (what’s happening under the hood) • 97	
Why don’t other LLMs behave like this? • 98	
General tips and advice for prompting DeepSeek 99	
The few-shot fallacy • 99	

System prompts • 100	
The verbose prompt trap • 100	
Other factors impacting DeepSeek’s response to prompts • 103	
Advanced techniques and tooling for structured output	109
Native JSON mode • 111	
Function calling • 112	
Type-enforced generation with Pydantic (via Instructor) • 114	
<i>Using Instructor for type safety</i> • 114	
Strategies for robustness and special cases • 116	
V series: unique prompting techniques for V-series models	121
The template tango: Unicode characters and special tokens • 121	
The formatting fiesta: When Markdown goes wild • 123	
The context window confusion • 124	
Hidden superpowers: Features the R series doesn’t have • 126	
Troubleshooting	129
Prompt migration guide	133
Summary	138

Part 2: Using DeepSeek 141

Chapter 4: Using DeepSeek: Case Studies 143

Technical requirements	144
Setting up your development environment • 144	
Configuring DeepSeek in the Cursor IDE • 145	
Organizing your development workspace • 146	
Configuration management • 147	
Benchmarking tools setup and prompt design	148
Use case study: Document understanding	150
Prompting DeepSeek • 150	
Response evaluation • 151	

<i>Evaluation methodology for technical analysis</i>	• 155
Follow-up code generation request	• 160
Follow-up response evaluation	• 166
<i>Evaluation methodology and metrics selection</i>	• 166
Recalibration through iterative prompting	• 169
<i>Practice exercise</i>	• 170
Use case study: Financial document analysis and benchmarking	170
Test document creation and benchmarking setup	• 171
Prompting DeepSeek for financial document extraction	• 174
Response evaluation	• 176
<i>Metric 1: Field extraction accuracy</i>	• 176
<i>Metric 2: Table parsing accuracy</i>	• 177
<i>Metric 3: Entity recognition accuracy</i>	• 177
<i>Metric 4: Structure preservation score</i>	• 178
<i>Metric 5: Processing time</i>	• 179
<i>Metric 6: Cost per document</i>	• 179
Test data analysis	• 180
Computing the final score	• 182
Benchmark comparison: From document to accuracy scores	183
Understanding DeepSeek-R1's extraction errors	• 187
Comparing the three approaches	• 188
<i>Testing Docling and MarkItDown</i>	• 188
<i>Computing benchmark metrics</i>	• 189
<i>Quantitative comparison across six metrics</i>	• 190
<i>Interpreting the results</i>	• 191
<i>Understanding tool capabilities</i>	• 192
<i>Choosing the right tool for your use case</i>	• 193
Hybrid approach: Best practices	• 194
<i>Recommended workflow architecture</i>	• 194
<i>Cost-benefit analysis and economic justification</i>	• 195
Summary	196

Technical requirements	201
Building the first prototype	202
Fetching our data • 202	
Creating the context • 207	
Defining the structured output • 211	
Creating the Daily Health Summary • 212	
Refactoring into an API • 216	
Deploying with Docker • 220	
Interacting with DeepSeek models	222
LiteLLM • 222	
Running locally with Ollama • 224	
CPU-based inference with Transformers and XGrammar • 227	
Refactoring for local generation on the CPU • 231	
Deploying an isolated model service with AWS	233
Inference backends • 235	
Deploying DeepSeek with LMI containers • 237	
Updating our service to use Amazon SageMaker endpoints • 245	
Best practices and recommendations	246
Summary	246

Chapter 6: Agents with DeepSeek

Technical requirements	250
A gentle introduction to agents	250
Tools	251
Understanding the Model Context Protocol	254
Working with agents and workflows	257
Exploring various agentic systems	260
Workflow: Evaluator-optimizer • 260	
An example: Summarizing arXiv papers • 261	

Workflow: Orchestrator-workers • 270	
<i>An example: A report-generating workflow • 271</i>	
Agent: Tool-calling agent • 279	
<i>An example: A web search agent • 280</i>	
<i>An important note about evaluating agents • 288</i>	
Summary	289
Part 3: Distilling and Deploying DeepSeek	291
Chapter 7: DeepSeek-Driven Fine-Tuning of Gemma 3 for Legal Reasoning	293
Technical requirements	294
Dependencies • 294	
Creating your local environment with ZenML • 294	
<i>Creating your ZenML Cloud account • 294</i>	
<i>API keys and environment variables • 295</i>	
Enhanced CUAD dataset • 296	
Fine-tuning without ZenML (standalone script) • 296	
<i>Optional standalone script to fine-tune without ZenML • 296</i>	
Understanding the importance of distillation and fine-tuning	297
Use case and dataset • 301	
The multi-label extraction problem in legal texts • 301	
Introducing CUAD: A structured benchmark for legal clause classification	302
Extending CUAD: Why we add rationales • 303	
Overview of the distillation fine-tuning process with CUAD and enhanced CUAD datasets • 303	
LLMOps tools for model distillation	304
The two-stage workflow for legal rationale distillation	308
Stage 1: Distillation • 308	
ZenML pipeline data processing • 314	
<i>The NONE label decision • 315</i>	

Instructional format for fine-tuning • 317	
Stage 2: Fine-tuning Gemma 3 on CUAD • 318	
<i>The fine-tuning process</i> • 319	
<i>Training dynamics of legal AI learning</i> • 323	
Evaluation and results	324
Performance metrics • 325	
Evaluation results • 326	
Error analysis: Understanding model limitations • 326	
Performance optimization potential • 329	
Key takeaways	331
When does distillation and fine-tuning make sense? • 331	
Summary	332
Chapter 8: Deploying DeepSeek Models	335
Technical requirements	336
The DeepSeek deployment landscape	336
DeepSeek's deployment quirks • 337	
Why self-deploy and what makes DeepSeek unique? • 337	
A decision-making framework for choosing your deployment strategy	341
Cost sanity check • 344	
Three paths to deployment • 346	
<i>Path 0: The baseline (official DeepSeek API)</i> • 347	
<i>Path 1: Local and on-premises (for dev and specialized cases)</i> • 348	
<i>Path 2: Managed inference services (the balanced approach)</i> • 350	
<i>Path 3: DIY on IaaS (for maximum control)</i> • 351	
Hardware and inference optimization engines for deployment	354
Choosing your hardware • 354	
Inference engines • 355	
The power of quantization • 357	
Hands-on deployment guides	359
Example 1: Local deployment with Ollama and a quantized DeepSeek Coder model • 361	

<i>Prerequisites and performance expectations</i> • 361	
<i>What this teaches</i> • 363	
Example 2: Managed deployment on Amazon Bedrock • 363	
<i>What this teaches</i> • 365	
Example 3: Deployment of DeepSeek V3 to the cloud using Hugging Face Inference	
Endpoints • 366	
<i>What this teaches</i> • 374	
Production operations and monitoring	374
Monitoring and observability • 375	
Scaling and performance • 376	
Cost management • 379	
Security in practice • 381	
CI/CD for models • 382	
Your deployment playbook	385
Summary	387
Chapter 9: Epilogue	389
<hr/>	
Chapter 10: Appendix	391
<hr/>	
Technical requirements	391
Getting started with the official DeepSeek API	392
Setting up • 392	
<i>Using the available models</i> • 397	
<i>Temperature</i> • 397	
<i>Pricing and rate limits</i> • 398	
API features • 399	
<i>Reasoning</i> • 399	
<i>Streaming</i> • 400	
<i>JSON output</i> • 400	
<i>Function calling</i> • 401	
FIM • 402	

Using common third-party APIs	402
Cloudflare • 403	
<i>Quirks and tips</i> • 406	
AWS • 406	
<i>Quirks and tips</i> • 410	
OpenRouter • 411	
<i>Quirks and tips</i> • 413	
Working with Cursor’s IDE for DeepSeek	414
Setting up your development environment • 414	
Configuring DeepSeek in Cursor’s IDE • 414	
<i>Direct integration: Connecting Cursor to DeepSeek</i> • 415	
<i>Alternative: Command-line integration</i> • 416	
Running or deploying DeepSeek yourself	417
Using llama.cpp • 417	
Ollama • 421	
Deploying DeepSeek yourself • 423	
Building your own setup for DeepSeek	424
LiteLLM • 424	
LangChain • 426	
Instructor • 428	
Other interesting libraries and resources • 430	
<i>Get This Book’s PDF Version and Exclusive Extras</i> • 431	
Chapter 11: Unlock Your Exclusive Benefits	433
Unlock this Book’s Free Benefits in 3 Easy Steps	433
Other Books You May Enjoy	439
Index	443