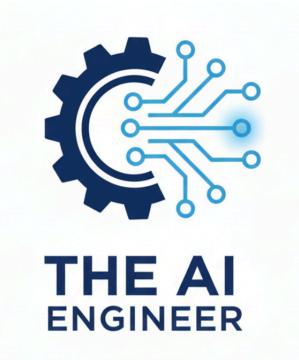
Deep Learning Basics with PyTorch

An approachable, code-first introduction to deep learning

Dr. Yves J. Hilpisch[1]

February 18, 2026



[1] Get in touch: https://linktr.ee/dyjh. Web page: https://theaiengineer.dev. Research, structuring, drafting, and visualizations were assisted by GPT 5.x as a co-writing tool under human direction. Comments and feedback are welcome.

Preface

Deep learning has moved from research labs into everyday products. From recommendation systems to generative models like GPTs, diffusion image generators, and multimodal assistants, modern AI is powered by deep neural networks. This book gives you a clear, code-first path into the field using PyTorch — a dominant framework in research and widely used in industry for NLP, LLMs, and beyond.

Why Deep Learning, Why Now

Classical machine learning remains essential, but many real-world problems demand flexible models that learn rich representations directly from data. Deep learning delivers:

- End-to-end learning: fewer hand-crafted features, more learning from raw inputs.
- Scale: models improve with more data, parameters, and compute.
- Transfer: pretrained models (e.g., BERT, GPT) can be adapted efficiently to new tasks.

The "transformer era" accelerated this trend. Attention mechanisms and large-scale training unlocked breakthroughs in language (GPT-style LLMs), vision (ViTs, diffusion), audio, and cross-modal reasoning. Understanding these foundations prepares you to contribute and to apply them responsibly.

Why PyTorch

PyTorch combines a clean, Pythonic API with eager execution and mature tooling. It is ubiquitous in:

- NLP and LLMs: most open-source transformer stacks and training recipes target PyTorch first.
- Research: dynamic graphs and strong ecosystem support rapid iteration and reproducibility.
- Production: TorchScript, ONNX export, and optimized runtimes make deployment practical.

We use PyTorch throughout to keep code concise, readable, and close to the math.

What You'll Learn (At a Glance)

This book is organized into five parts, each building on the last, plus practical appendices.

- Part I — data → features → models → metrics; limits of classical approaches.
- Part II — tensors, autograd, layers, activations, clean training loops with `nn.Module`.

- Part III — input pipelines, regularization, LR schedules, CNNs, and robust training at scale.
- Part IV — sequences and embeddings, attention and transformers, and reliable large-model training (DDP, AMP, checkpoints).
- Part V — ethics, risk, documentation, governance, and a learning path to keep growing.

Appendices provide quick references and runnable resources:

- Appendix A — Python & NumPy.
- Appendix B — Probability and Statistics.
- Appendix C — Linear Algebra.
- Appendix D — Calculus.
- Appendix E — Installation & Environment.
- Appendix F — Full Scripts.
- Appendix G — Notebooks Index.
- Appendix H — Glossary.

How to Use This Book

Every chapter is paired with runnable code and figures that you can reproduce.

- Code-first: every concept is paired with a minimal, runnable example. Figures are reproducible and generated from scripts in `code/figures/`.
- Compact math: just enough notation to reason about shapes, derivatives, and objectives — always tied back to code.
- Visual feedback: decision boundaries, loss curves, and feature maps build intuition at each step.

Who This Book Is For

Python programmers who want a clear path from ML fundamentals to modern deep learning with PyTorch. If you know basic Python and NumPy, you're ready. A refresher is provided in Appendix A.

Setup and Resources

This book supports both local and cloud workflows.

- Local or Colab: Appendix E walks through both. A GPU is optional for most early chapters, but helpful later for CNNs and transformer demos.
- Companion repository: provides the runnable code and notebooks with a simple layout — `code/` (scripts), `figures/` (outputs), `notebooks/` (experiments).
- Notebooks and full scripts: see Appendix F and Appendix G for complete listings and links.

Companion Repository

Browse the code and notebooks online or clone locally:

- Code and notebooks: github.com/yhilpisch/dlcode
- Structure: `code/` (chapter scripts and figure generators), `notebooks/` (interactive notebooks), `figures/` (generated images)

The rendered book (HTML/PDF) stands alone. The companion repository mirrors the structure so you can run, modify, and extend the examples at your own pace.

A Note on Scope

We emphasize supervised learning, practical training workflows, and the transformer foundations behind today's LLMs. The field is broad — reinforcement learning, self-supervised pretraining, and retrieval-augmented generation are noted where relevant and included in the "Next Steps" chapter for further study.

Thanks

This project stands on the shoulders of the PyTorch community and open-source contributors who share code, papers, and tools. Special thanks to readers who report issues and suggest improvements — the book is better because of you. Enjoy the journey, and build something you can show.

Technical & Legal Note

Contents

List of Figures

Contact

Deep Learning Basics with PyTorch

The AI Engineer

Get in touch:

https://linktr.ee/dyjh

https://theaiengineer.dev