Extended Mathematical Reading in Deep Learning

Mathematics of Deep Learning — Chapter 22

Yin Yang

Foundation Books Project

Roadmap

Why a Reading List

Objects Anchoring the List

Approximation, Optimization, Generalization

Geometry, Generation, Decisions

Reading Habits

Summary

Why a Reading List

What This Chapter Adds

The technical chapters built a working vocabulary. They also left several large topics open on purpose.

- This chapter is a curated entry point into those topics.
- Not a survey; not a curriculum; a pointer set with a usage rule.
- Designed to be re-read after every applied project.

A reading list is only useful if it can change what you do next Monday.

Four Questions Left Open

Even after the technical chapters, four questions remain genuinely hard:

- Why do some architectures express certain functions efficiently?
- Why does training find one minimum rather than another?
- How do classical generalization bounds connect to modern, overparameterized training?
- How should generative or RL systems be evaluated when the output is not one label?

Each question maps to a specialized field — and a path in this reading list.

Objects Anchoring the List

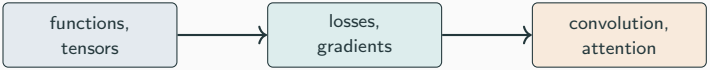
Objects the Book Already Gave You

functions,
tensors

Objects the Book Already Gave You



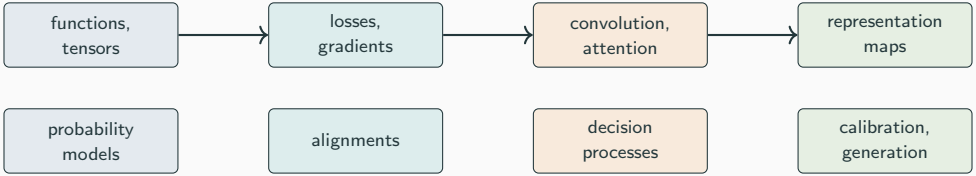
Objects the Book Already Gave You



Objects the Book Already Gave You



Objects the Book Already Gave You



Eight durable objects. Every reference below sits on at least one of them.

The Usage Rule

The list is meant to be *used*, not skimmed.

1. Pick one path.
2. Name the applied question it would change.
3. Identify the mathematical object: f_θ, optimizer, $\widehat{R}$, geometry, distribution, or policy.
4. Choose one reference.
5. Ask: smallest example? main assumption? failure mode?

A claim becomes useful only when it changes a concrete decision.

Approximation, Optimization, Generalization

Path 1: Approximation and Architecture

Why some architectures express some functions efficiently.

- **Universal approximation:** broad classes are *representable*, but representability is not training.
- **Depth separation:** certain functions need depth, not just width, to be efficient.
- **Equivariant networks:** symmetries baked in (CNN, group equivariance).
- **Attention:** long-range mixing as a primitive.

Representable and trainable are different claims.

Path 2: Optimization and Implicit Bias

Many parameter settings fit the same data. Training picks one.

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \widehat{R}(\theta_t).$$

- **Stochastic optimization:** minibatch SGD, Adam, weight decay are not cosmetic details.
- **Implicit bias:** which fitting solution the trajectory prefers among many.
- **Sharpness / flatness:** useful diagnostic words — but reparameterization-sensitive, handle with caveats.

Recipe choices select between many equally-good fits.

Path 3: Generalization, Stability, Scaling

Low training loss is not enough once deployment matters.

$$\widehat{R}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f_\theta(x_i), y_i) \overset{?}{\approx} \mathbb{E}_{(X,Y)\sim P}[\ell(f_\theta(X), Y)] = R(\theta).$$

- Expected risk, empirical risk, validation estimates, distribution shift.
- **Algorithmic stability:** sensitivity of $\widehat{\theta}$ to a single training example.
- **Calibration:** confidence and correctness must agree where decisions are made.

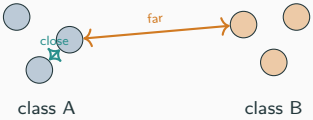
The deployment gap is a mathematical object, not a vibe.

Geometry, Generation, Decisions

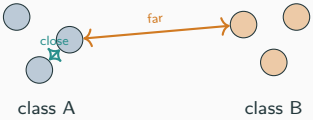
Path 4: Representation Geometry and Transfer



Path 4: Representation Geometry and Transfer



Path 4: Representation Geometry and Transfer



Contrastive objectives: matched pairs close, unmatched pairs far. Transfer asks whether this geometry survives a new task.

Applied diagnostic: are examples that should behave similarly actually close?

Path 5: Probabilistic and Generative Modeling

When the model must produce data, the object is a distribution.

- **VAEs:** latent variables, ELBO, amortized inference.
- **GANs:** adversarial divergence minimization.
- **Score / diffusion:** learn $\nabla_x \log p(x)$, generate by reversing noise.

Evaluation can disagree across criteria:

Criterion	What it scores
Likelihood	probability mass on real data
Sample quality	realism of generated draws
Diversity	coverage of the data distribution
Usefulness	downstream task performance

Name which property matters before reporting a number.

Path 6: Sequential Decisions, Preferences, Trust

When decisions feed back into the data they shape.

- **Markov decision processes:** policies π, rewards r, values V, returns G.
- **Policy optimization:** REINFORCE, PPO — gradient methods on a non-supervised signal.
- **Preference learning:** comparisons replace labels (RLHF, DPO, GRPO).
- **Robust evaluation:** distribution shift, confident errors, optimizing against the wrong signal.

Pick the object first — distribution, policy, comparison — then the algorithm.

Reading Habits

Strong Candidates for Deeper Study

A topic is worth a sustained reading project when it has:

- a clear applied task

Strong Candidates for Deeper Study

A topic is worth a sustained reading project when it has:

- a clear applied task
- a durable mathematical object

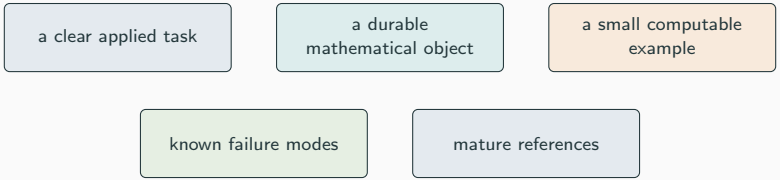
Strong Candidates for Deeper Study

A topic is worth a sustained reading project when it has:

- a clear applied task
- a durable mathematical object
- a small computable example

Strong Candidates for Deeper Study

A topic is worth a sustained reading project when it has:



All five together; missing any one usually means “not yet ready to teach.”

Reading Checks: Four Questions Per Topic

For any one topic, a useful reading note should answer:

1. What applied decision would this idea change?
2. What object: function class, optimizer, risk estimate, representation geometry, distribution, or policy?
3. What is the smallest numerical or symbolic example?
4. What assumption or failure mode would make the result misleading in an applied system?

Four questions, every topic, every time.

Questions That Outlast Model Names

Architecture names change. The durable questions remain.

- What function is being learned?
- What loss is being optimized?
- What assumptions connect training evidence to deployment use?
- What representation geometry is useful?
- What would count as failure?

These questions should outlast any single model name.

Summary

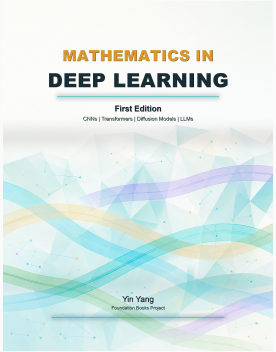
Chapter in One Slide

- **Purpose:** pointer set, not a survey; usable after every applied project.
- **Anchors:** eight objects the book already named — functions, tensors, losses, gradients, convolution, attention, representations, probability, alignments, MDPs, calibration, generation.
- **Six paths:** approximation, optimization, generalization, representation geometry, generative modeling, sequential decisions.
- **Usage rule:** pick a path, name the question, identify the object, choose one reference, ask the four checks.
- **Durable questions:** learned function, optimized loss, training-to-deployment bridge, useful geometry, definition of failure.

Object first, algorithm second.

One question, one reference, one example.

A list is only useful when it changes a decision.



Mathematics in Deep Learning

Chapter overview slides for the book by Yin Yang.

Book page	`foundation-books.github.io/mathematics-in-deep-learning`
Code	`github.com/foundation-books/mathematics-in-deep-learning-code`
Paperback	`amazon.com/dp/B0H1ZZHDT2`
Hardcover	`amazon.com/dp/B0H1ZL11MJ`
Kindle	`amazon.com/dp/B0GX32M8SP`
License	CC BY-NC-ND 4.0

These free overview slides may be shared non-commercially with attribution and without modifications. Full explanations, exercises, and companion-code context are in the book and linked repository.