Lesson 1.5 — Logarithms: Extracting Hidden Scaling

Mathematical Intuition · Module 1 — Operations

Node 1.5

1. Why this matters (Hook)

If exponentiation is **scaling-of-scaling**,
then logarithms are the **decoder**.

A logarithm answers one simple question:

"How many times was I scaled?"

That is why logs appear everywhere:

- information theory
- entropy
- signal processing
- machine learning (log-likelihoods, cross-entropy)
- statistics (log-odds, logit)
- quantum mechanics (log-amplitude)
- complexity theory
- gradient methods
- normalization and stability
- fractals and dimensionality

Logs let you *see inside* a multiplicative structure.

They turn multiplicative complexity into additive clarity.

And that is their superpower.

---

2. Core Intuition

2.1 Exponentials: build scaling

$$a^x = \underbrace{a \times a \times \cdots \times a}_{x \text{ times}}$$

2.2 Logarithms: reveal scaling

$$x = \log_a(y)$$

Which means:

"y was produced by scaling a by itself x times."

Logarithms invert the exponential *conceptually*, not just algebraically.

They extract:

- repeated scaling
- growth depth
- information content
- contrast in data
- hidden multipliers
- order of magnitude

2.3 The Core Sentence

Logarithm =
"What power produced this result?"

But deeper:

Logarithm = "How deep into the scaling hierarchy are we?"

---

3. What Logs *Actually* Do

3.1 Turn multiplication into addition

$$\log(ab) = \log a + \log b$$

Not a trick.

A structural truth:

> **Multiplication combines scalings.**
> **Logarithms combine *counts of scalings*.**

This is the reason logs dominate probability, ML, and information theory.

Multiplicative processes → logs give linear, additive structure.

3.2 Extract rate from exponential growth

If:

$$y = e^{kt},$$

then:

$$\ln y = kt.$$

The logarithm removes the exponential wrapping and returns the generator.

Logs expose:

- growth rate
- decay constant
- eigenvalue
- depth
- number of compounding steps

3.3 Reveal hidden order of magnitude

The difference between:

- 10
- 100
- 1000

is invisible under addition
but perfectly visible under log:

- $\log 10 = 1$
- $\log 100 = 2$
- $\log 1000 = 3$

Logs see *scale*, not magnitude.

3.4 Convert products of probabilities into sums

In ML, the likelihood of N independent events:

$$P = p_1 p_2 \cdots p_n$$

becomes:

$$\log P = \log p_1 + \log p_2 + \cdots + \log p_n.$$

This turns **numerical collapse** into **numerical stability**
and **multiplicative complexity** into **additive clarity**.

3.5 Reveal dimensionality / fractal structure

Fractal dimension:

$$D = \frac{\log N}{\log(1/r)}$$

Log extracts the scaling pattern that defines the structure.

---

4. Physical & Mathematical Interpretations

4.1 Entropy (information per uncertainty)

Shannon entropy:

$$H = -\sum p_i \log p_i$$

Interpretation:

> **log p tells you how "surprising" or "compressed" an event is. Multiplying by p weights by likelihood.**

Logs quantify informational "depth."

4.2 Sound: decibels

$$\text{dB} = 10 \log_{10}\left(\frac{I}{I_0}\right)$$

Interpretation:

> "How many orders of magnitude stronger is this intensity compared to baseline?"

Sound perception is logarithmic because human ears are adapted to ratios, not differences.

4.3 Quantum mechanics: log-amplitude

If:

$$|\psi|^2 = e^{-\alpha x^2}$$

then:

$$\log |\psi|^2 = -\alpha x^2$$

reveals:

- underlying Gaussian structure
- potential shape
- symmetry

Logs simplify wavefunctions into linear forms.

4.4 ML: log-odds & logistic regression

Logit:

$$\operatorname{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

Interpretation:

> "How many log-units does the probability favor success vs failure?"

This converts probabilities into a linear space where gradient descent works.

4.5 Complexity theory

$$O(\log n)$$

means:

> "Work increases by one unit for each doubling of n."

Logarithms measure depth of recursive decomposition.

---

5. Cross-Domain Examples

5.1 Physics — half-life

Decay:

$$N(t) = N_0 e^{-kt}$$

Taking logs:

$$\ln N(t) = \ln N_0 - kt$$

Log removes exponent — reveals linear decay in log-space.

5.2 Neural networks — cross entropy

$$L = -\sum y_i \log p_i$$

Each log measures the "penalty depth" of assigning low probability to the true class.

5.3 Compression — code length

If an event has probability p, its optimal code length is:

$$\ell = -\log_2 p$$

Meaning:

> "Rare events require more bits."

5.4 VQE — log of gradients (advanced intuition)

Gradients often behave exponentially depending on ansatz depth.

Taking logs reveals:

- sensitivity
- collapse
- explosion
- relative scaling

Logs let us diagnose landscapes.

---

6. Micro Drills

Answer conceptually.

1. What does a logarithm measure?
   Give the shortest correct intuition.
2. Interpret:

   $$\log(ab) = \log a + \log b.$$

   What structural truth does this express?
3. Interpret:

   $$\ln y = kt$$

   in the context of exponential growth.
4. Why do ML practitioners always take logs of probabilities?
5. What does

   $$-\log p$$

   measure in information theory?
6. Why are human senses (sound, brightness) often logarithmic?

---

7. Reflection

Pick one instance where logs appear in your work:

- loss functions
- entropy
- exponential decay
- normalization
- softmax stability
- quantum wavefunctions
- VQE gradient behavior

Write 2–3 sentences explaining:

- what scaling process the log is *inverting*
- what hidden structure is being revealed
- why addition in log-space is easier than multiplication in real-space

Logs are the bridge between scaling and understanding.

Mathematical Intuition · Module 1 — Operations · Node 1.5
Concepts: shannon-entropy, likelihood, information-theory

hbar.university — own.your.cognition