The Monad: A Formal Model of the Learning Process

Radu Bucea-Manea-Tonis[1]

Abstract: Objectives: This paper aims to apply categorical math notions and functional programming in the cognitive field of study. **Prior Work:** I am using a monad, an endofunctor in the category of monoids, to represent the process of learning. **Approach:** For this purpose, I have used financial math modeling and Taylor series. **Results:** The model obtained combines compound interest rate with structural correctors and tail-decaying order terms. **Implications:** This paper is important for employing methods from different fields of study (finance, math, functional programming) to develop a versatile model for the learning process. **Value:** This paper is original for applying natural growing metrics and the auto-similarity concept in cognitive sciences.

Keywords: monad; Euler's number; auto-similarity; Taylor series; decay order

1. Introduction

There is immense progress being made in the cognitive sciences with the advent of AI, but there is no consensus on whether mind processes are quantifiable or even computable at all (Koellner, 2018). This paper provides a model to measure learning by calculating the rate of learning, forgetting, and means of correcting any structural or external biases in this process.

[1] Senior lecturer, Ph.D., Danubius International University of Galati, Romania, Address: 3 Galati Blvd., Galati 800654, Romania, Corresponding author: radumanea@univ-danubius.ro.

Copyright: © 2026 by the authors.
Open access publication under the terms and conditions of the
Creative Commons Attribution-NonCommercial (CC BY NC) license
(https://creativecommons.org/licenses/by-nc/4.0/)

Being a naturally paced growing process, I imagined the mind operating like a monad structure on two steps:

$$unit: A \to M(A)$$

$$bind: M(A) \to (A \to M(B)) \to M(B)$$

First, the Unit will identify any knowledge *A* worth being lifted by mapping using a functor specially designed for this purpose *M* - map. Secondly, the knowledge being mapped is integrated into the mind context as valid and compatible knowledge, then flattened (flatmap) into a new, improved mind state. The transition between states is commanded by *A → M(B)* directive of the Bind step, with the meaning of extracting *A* and integrating it into a different context *M(B)*.

Being modelled as a monad, mind should also be governed by monad rules as follows:

$$bind(unit(x), f) = f(x)$$

$$bind(m, unit) = m$$

$$bind(bind(m, f), g) = bind(m, x \mapsto bind(f(x), g))$$

Basically, mind seems to work according to composable laws, every new state of mind resulting from self-applying the same sequence (e.g. bind & unit) on different kinds of knowledge. This continuous integration of new knowledge with the existing one, might be responsible for consciousness or a self-aware mindset, triggered by the symbolic interference of the internal level of knowledge with the external state of affairs, according to the being-in-the-world (germ. *dasein*) concept (Heidegger, 1967).

The following diagrams are used interchangeably to represent monads:

$$A \to^{unit} M(A)$$

$$M(A) \to^{bind(f)} M(B)$$

2. Auto-Similarity Concept Applied in the Learning Process

Knowledge is accumulated by a continuous process described by a rate of learning. The next expression describes a process where each step adds a tiny fraction $^{1}/_{x}$ of progress, repeated *x* times. Each update is multiplicative:

new state = old state × (1+ small improvement)

The limit is classical and well established in calculus for $x \to \infty$; please see Figure 1:

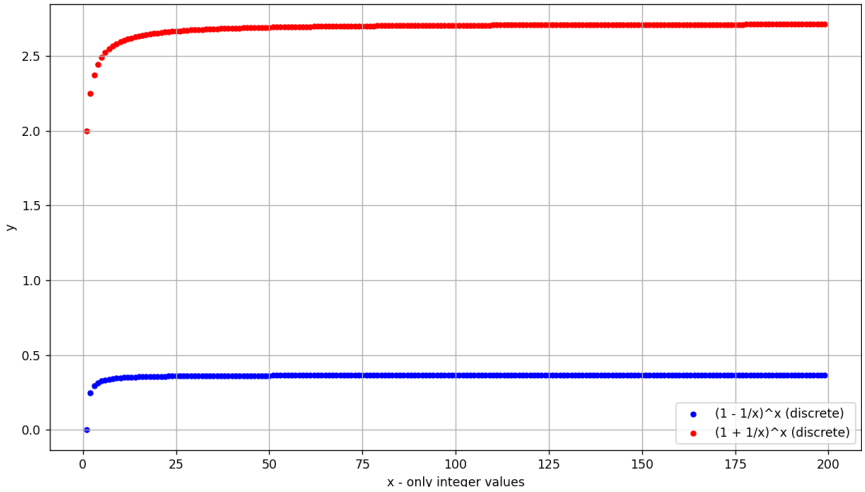


Figure 1. Graphic representations of $(1 \pm {}^{1}/_{x})^{x}$ functions.

Source: Created by author in Python, according to Annex 1

According to Mandelbrot (1982), we're able to define the core autosimilar function based on the speed of learning, which must be proportional to what has already been learned.

So, for

$$f(x) = \left(1 + \frac{1}{x}\right)^{x} = \left(\frac{x+1}{x}\right)^{x}, \qquad x \neq 0.$$

, and its logarithmic representation:

$$lnf(x) = xln\left(1 + \frac{1}{x}\right) = xln\left(\frac{x+1}{x}\right).$$

I'm using logarithmic differentiation to obtain its derivative:

$$\frac{f'(x)}{f(x)} = ln\left(1 + \frac{1}{x}\right) - \frac{1}{x+1}.$$

Thus

$$f'(x) = \left(1 + \frac{1}{x}\right)^{x}\left[ln\left(1 + \frac{1}{x}\right) - \frac{1}{x+1}\right].$$

In conclusion, the derivative is:

f'(x) = f(x) × (some factor depending on x)

3. Building Asymptotic Taylor Series as $x \to \infty$

I use a Taylor-type asymptotic expansion with controlled decay of remainder terms, according to (Boyd, 1999):

$$ln\left(1+\frac{1}{x}\right)=\frac{1}{x}-\frac{1}{2x^2}+\frac{1}{3x^3}-\frac{1}{4x^4}+O\left(\frac{1}{x^5}\right).$$

Then

$$xln\left(1+\frac{1}{x}\right)=1-\frac{1}{2x}+\frac{1}{3x^2}-\frac{1}{4x^3}+O\left(\frac{1}{x^4}\right).$$

Thus

$$f(x)=e^{\,1-\frac{1}{2x}+\frac{1}{3x^2}+O(1/x^3)}=e\cdot exp\left(-\frac{1}{2x}+\frac{1}{3x^2}+O\left(\frac{1}{x^3}\right)\right).$$

, and by expanding the exponential:

$$f(x)=e\left(1-\frac{1}{2x}+\frac{11}{24x^2}+O\left(\frac{1}{x^3}\right)\right).$$

We separate the ideal truncated model from two symbolic error terms to obtain forgetting and learning errors:

$$f(x)\approx e\left(1-\frac{1}{2x}+\frac{11}{24x^2}\right)-\varepsilon_{forget}(x)+\varepsilon_{learn}(x).$$

Now, the forgetting error captures the tail of the omitted higher-order terms:

$$\varepsilon_{forget}(x)=e\cdot O\left(\frac{1}{x^3}\right).$$

Thus, the learning error captures any adaptive, empirical, or context-driven correction beyond the pure series, like:

$$ln\left(\frac{x+1}{x}\right)-\frac{1}{x+1}=\frac{(x+1)\,ln\left(\frac{x+1}{x}\right)-1}{x+1}.$$

4. Conclusion

Both learning and forgetting functions belong to the same autosimilar family:

$$\left(1+\frac{c}{x}\right)^{x},$$

with

$$c = 1 \quad (learning), \qquad c = -a \quad (error\ rate).$$

Their limits are

$$\left(1+\frac{1}{x}\right)^{x} \to e, \qquad \left(1-\frac{a}{x}\right)^{x} \to e^{-a}.$$

Their expansions differ only by the parameter c, and thus the forgetting error function must have the same Big–O structure:

$$\varepsilon_{forget}(x) = e^{c} O\left(\frac{1}{x^{3}}\right).$$

References

Boyd, J. P. (1999). The asymptotic Taylor series and its remainder decay. *Applied Mathematics and Computation*, *105*(1), 79–94.

Heidegger, M. (1967). *Sein und Zeit* [Being and Time]. Niemeyer.

Koellner, P. (2018). On the Question of Whether the Mind Can Be Mechanized, II: Penrose's New Argument / On Whether the Mind Can Be Mechanized. *The Journal of Philosophy*, *115*(9), 453–484.

Mandelbrot, B. (1982). Self-similarity and fractional dimension. *Proceedings of the National Academy of Sciences*, *79*(10), 4054–4060.

Annex 1. Python source code for graphic representation of functions (D, G)

```
import numpy as np
import matplotlib.pyplot as plt

# x doar pt. valori întregi
x = np.arange(1, 200)

D = (1 - 1/x)**x
G = (1 + 1/x)**x

plt.figure(figsize=(10, 6))

# puncte discrete
plt.scatter(x, D, label="(1 - 1/x)^x (discrete)", color="blue", s=15)
plt.scatter(x, G, label="(1 + 1/x)^x (discrete)", color="red", s=15)

plt.title("Graphic represententions of (1 ± 1/x)^x functions")
plt.xlabel("x - only integer values")
plt.ylabel("y")
plt.grid(True)
plt.legend()
plt.tight_layout()

plt.show()
```