Whitepaper: The Persistence Stack

Contents

Whitepaper: Implementation of the Fractal Persistence Law

Introduction

This document outlines the technical implementation of the theoretical framework described in `docs/persistence.md`. The core objective is to transition the "Fractal Persistence Law"—which posits that existence is a non-equilibrium steady state requiring a sustainability ratio $\mathcal{R} \geq 1$—into a computable neural architecture.

The implementation in this repository is split into two lines:

- **Reference implementation (legacy):** a recursive graph of nodes where learning is driven not by global loss functions, but by the local drive for persistence (as described below).
- **Current implementation (canonical):** `nanochat/nanochat/fractal_gpt.py` + `nanochat/nanochat/fractal_` which implements persistence as per-token $\mathcal{R}$ modulation on top of a fast transformer trunk (FractalLM). This is the version used by the cross-project bridges (nanochat → ai3 → blockchain / social-media).

Mapping Theory to Implementation

The theoretical variables of the Fractal Persistence Equation (FPE) are mapped directly to the code in `src/glstm.py`:

1. The Sustainability Ratio ($\mathcal{R}$)

In the theory, $\mathcal{R}$ is the primary metric of existence. In the implementation, this is tracked in `LayerPersistenceState.R` as a tensor of ratios per node per layer. The condition $\mathcal{R} \geq 1$ is enforced via the `persistence_loss` function, which penalizes any state where $\mathcal{R} < 1$.

2. Model Divergence ($\mathcal{D}_{KL}$)

The theory describes $\mathcal{D}_{KL}$ as the "Delusion Penalty"—the divergence between the system's internal model and reality. This is implemented in `LayerPersistenceState.update_recursive_persistence` using Kullback-Leibler divergence:

```
d_kl = F.kl_div(pred.log(), target, reduction='none').sum(dim=-1, keepdim=True)
```

This value acts as the "entropy tax" in the denominator of the persistence calculation.

3. Fractal Dependency (Ψ, Φ)

The FPE asserts that persistence is a property of the graph, not the individual. The implementation realizes this through recursive dependencies:

- **Children Influence**: `children_R` is the mean persistence of the layer below.
- **Parent Influence**: `parents_R` is the mean persistence of the layer above.
- **Geometric Integration**: The `neighbor_R` (geometric mean of parents and children) ensures that a node cannot persist if its structural context has collapsed.

4. Energy Flow (P_{in})

The `energy_in` parameter in `update_recursive_persistence` represents the available computational power/energy flow. The persistence update formula is:

$$\mathcal{R} = \frac{\text{energy}_{\text{in}} \cdot \text{neighbor}_R}{1 + \mathcal{D}_{KL}}$$

Architectural Realization

Recursive Node Dynamics

The `RecursivePersistenceNode` uses an LSTM cell to maintain state, aggregating information from children via an attention mechanism and receiving top-down "persistence pressure" from its parent. This mirrors the biological and cybernetic structures described in the whitepaper.

The Persistence Loop

The system operates in two primary phases:

1. **Bottom-Up Pass**: Information flows from leaf nodes to the root, aggregating structural complexity.
2. **Top-Down Pass**: Persistence ratios are calculated and "pressure" signals are sent back down the graph.

Learning as Survival

Unlike traditional LLMs that use global backpropagation to minimize a cross-entropy loss, this architecture treats persistence as the learning signal. The `get_persistence_gradient` method generates "persistence pressure" when $\mathcal{R} < 1$, forcing the node to adapt its internal state to reduce $\mathcal{D}_{KL}$ and regain stability.

Conclusion

The implementation in `src/` transforms the Thermodynamics of Existence from a philosophical proof into a functional architectural constraint. By replacing global gradients with local persistence requirements, the system evolves as a fractal graph where the only surviving patterns are those that achieve "informational profit"—their predictive accuracy is sufficient to overcome the noise of their environment.