Member-only story

Efficient and Interpretable AI Models Through Sparse Nonlinearity

Manuel Brenner Following 9 min read · 1 hour ago

ML-driven sequence models are everywhere, underpinning chatbots like ChatGPT and Claude, AI weather forecasts, the 2024 Nobel Prize-winning work in physics (Hopfield networks) and chemistry (protein structure prediction), financial forecasting, and much more.

Sequence model learn patterns in ordered data by **identifying relationships between elements** across time or position. Different architectures solve this

differently: RNNs process sequentially with explicit memory, while Transformers use attention to relate all positions simultaneously.

The range of sequence modeling tasks, and corresponding architectures, is quite broad. A **many-to-one** task might classify a movie review as positive or negative after reading all the words. **One-to-many** tasks generate sequences from a single input, like forecasting the weather for the rest of the week based on the current weather, while **many-to-many** tasks handle tasks like language translation, where both input and output are sequences.

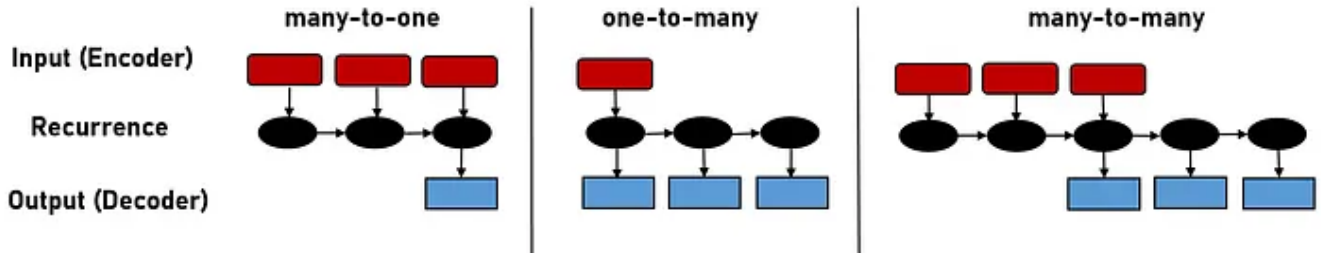


Image by the author.

Given the scope of applications of sequence models in science and beyond, understanding and optimizing these models is one of the most important research directions in modern ML.

A crucial lens for analyzing ML models in general is via the lense of **linearity versus nonlinearity**. In my previous article on "A Guide To Linearity and

Nonlinearity in Machine Learning", I went in depth into how different facets of nonlinearity manifest in ML.

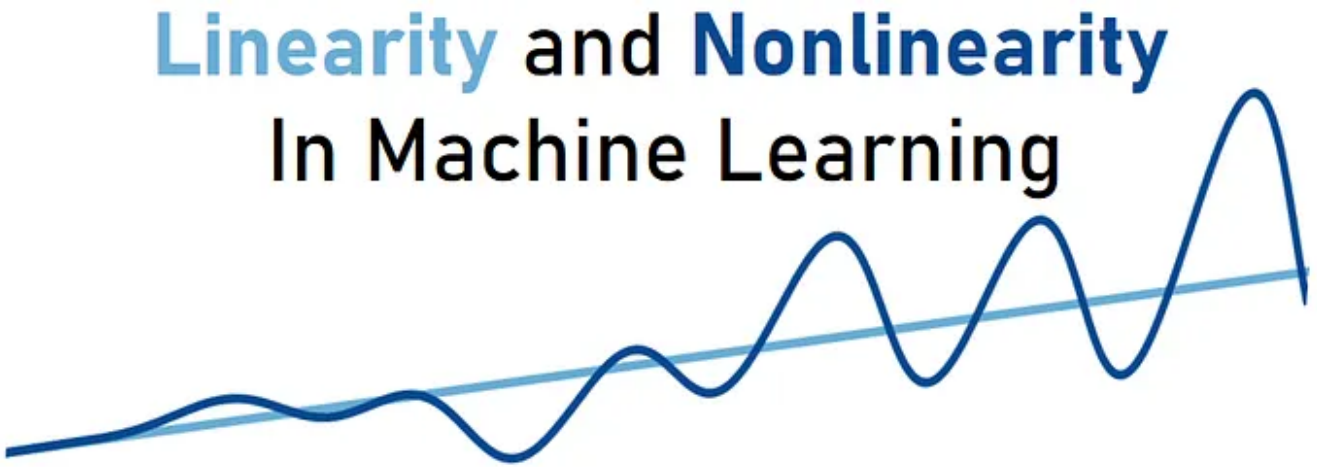


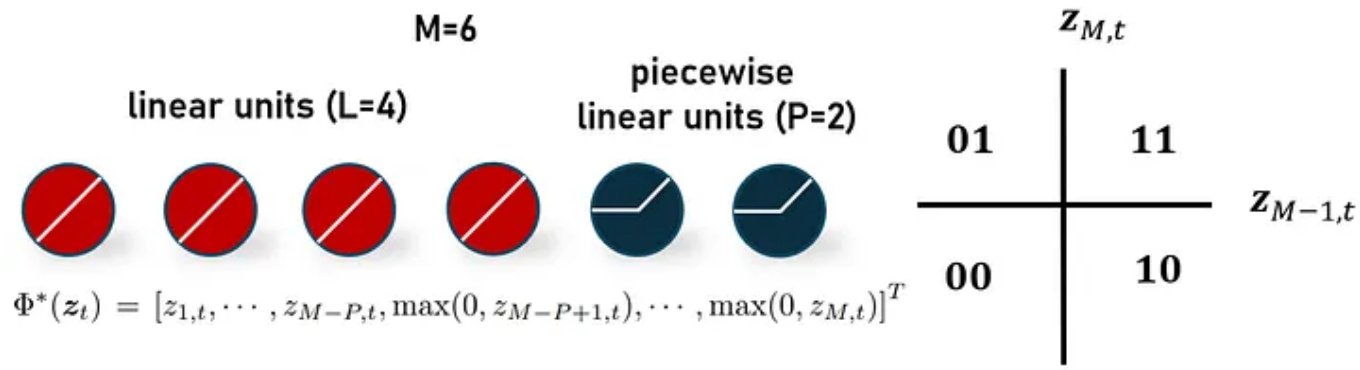
Image by the author.

The tradeoff is fundamental: linear models are easy to optimize and interpret but limited in expressivity. This matters especially for sequence modeling, where it directly affects training efficiency. Linear recurrence enables efficient parallel training, while nonlinearity is crucial for complex functions like contextual processing or chaos. As LLMs scale to ever-larger context windows, processing thousands of tokens simultaneously to generate better responses, the question of parallel training efficiency has literally become a billion-dollar problem. How much nonlinearity do we actually need, and where should it go?

Our novel paper "Uncovering the Computational Roles of Nonlinearity in Sequence Modeling Using AL-RNNs" which recently appeared in Transaction on Machine Learning Research (TMLR), explores this question of nonlinearity in sequence models in depth. In this article I want to explain its main findings.

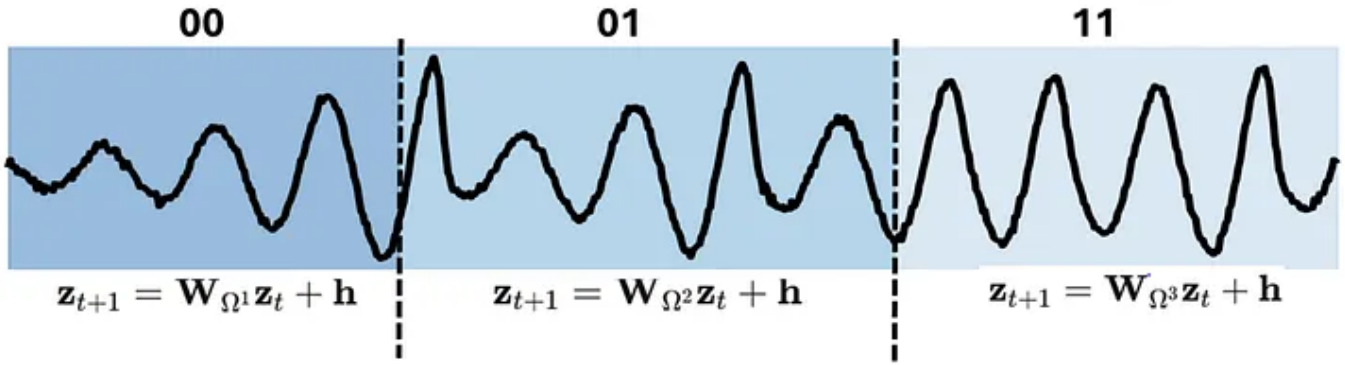
Many ML problems require nonlinearity as a functional necessity. The XOR function, for instance, can't be implemented by any linear network, no matter how large or deep. Nonlinearity enters neural networks through activation functions like the ReLU, tanh, softmax, that transform representations at each layer. The choice is usually binary: fully linear or fully nonlinear, and not much in between. This all-or-nothing approach works fine with abundant data when interpretability isn't a priority. But in scientific and practical settings where data is limited, finding the right level of complexity is crucial for discovering meaningful explanations.

In our 2024 NeurIPS paper on Almost-Linear recurrent neural networks (AL-RNNs), we explored this middle ground by introducing a simple architectural constraint: we built RNNs where only a small subset of units use ReLU nonlinearity, while the rest remain linear.

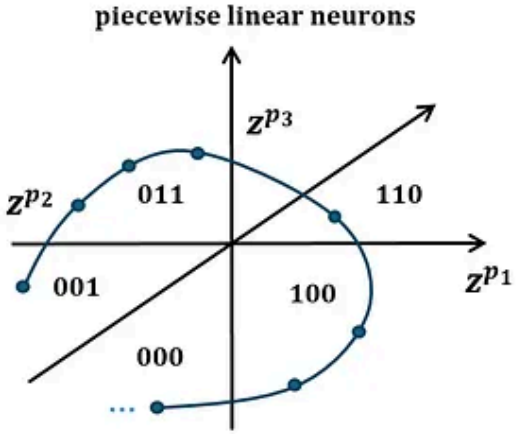


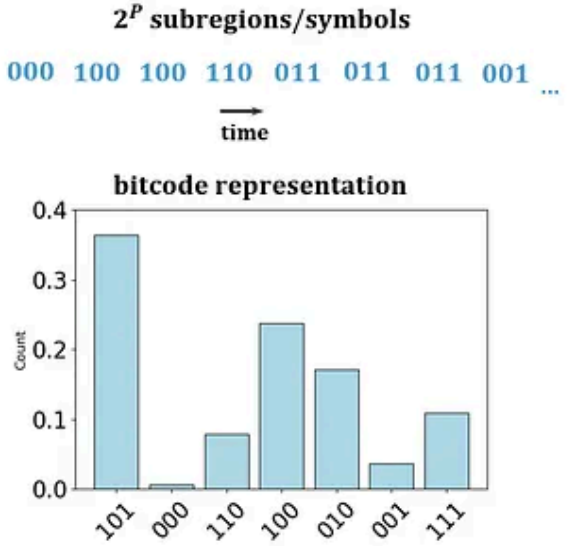
This setup has two main advantages. First, this design choice **decouples two aspects of model complexity that are usually intertwined.** The first is the size of the latent space (the total number of units M, hence the overall "size" or "capacity" of the model). The second is the degree of nonlinearity (how many units have a nonlinear activation function). In standard architectures, adding more units automatically adds more nonlinearity, and restricting nonlinearity means restricting total capacity. AL-RNNs let us study these dimensions independently.

Second, the ReLU activation has a particularly useful property: it partitions the model's latent space into **linear regions.** Each point in the AL-RNN's state space belongs to one such region. Intuitively, every trajectory traversed in the RNNs state space is split up into regions governed by a simple linear equation:

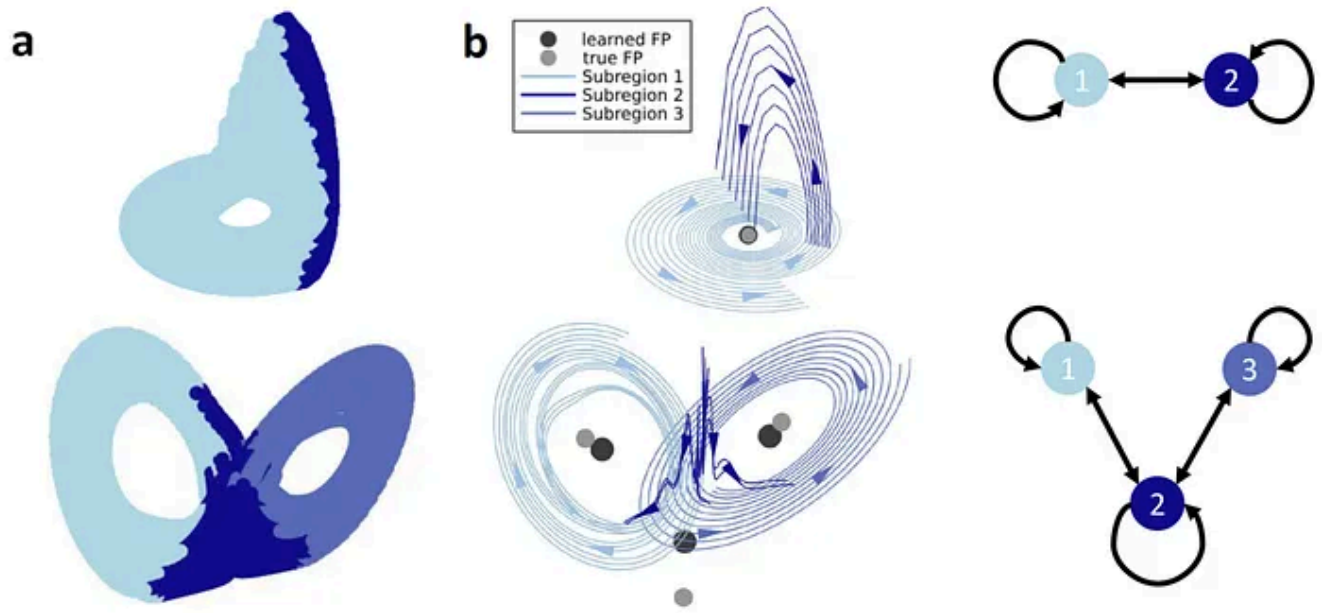


This naturally leads to what we call a “bitcode representation” across tasks that allows us to see across the entire training/test set in which linear subregions the model processes inputs:





In the NeurIPS paper, we explored a particularly challenging test case: **chaotic dynamical systems**. A chaotic attractor is a region of state space that the system's trajectory moves in complex, never-repeating patterns. The Lorenz attractor, for instance, resembles a butterfly with two lobes, where trajectories circle one lobe, then unpredictably switch to circle the other, creating the system's characteristic sensitive dependence on initial conditions (that you might be more familiar with in the context of the weather, and our inability to forecast it reliably for more than a couple of days). This complexity seems to demand sophisticated modeling approaches, and many ML papers have used more elaborate architectures like LSTMs, GRUs, or Transformers to reconstruct chaotic attractors. However, we found that AL-RNNs could capture the essential topological structure of these chaotic attractors using **just one or two ReLU units** in an otherwise linear network.



The Rössler attractor (top) and Lorenz-63 attractor (bottom) decompose into 2/3 distinct linear subregions (coloured in distinct hues of blue), with clear dynamical interpretations within each subregions.

On top, we found that the reconstructions were structured in a highly interpretable way. The model automatically partitions the attractor into distinct linear subregions, where within each subregion, the dynamics are purely linear, governed by a simple matrix equation, and complexity emerges from how trajectories transition between these regions as they cross ReLU thresholds.

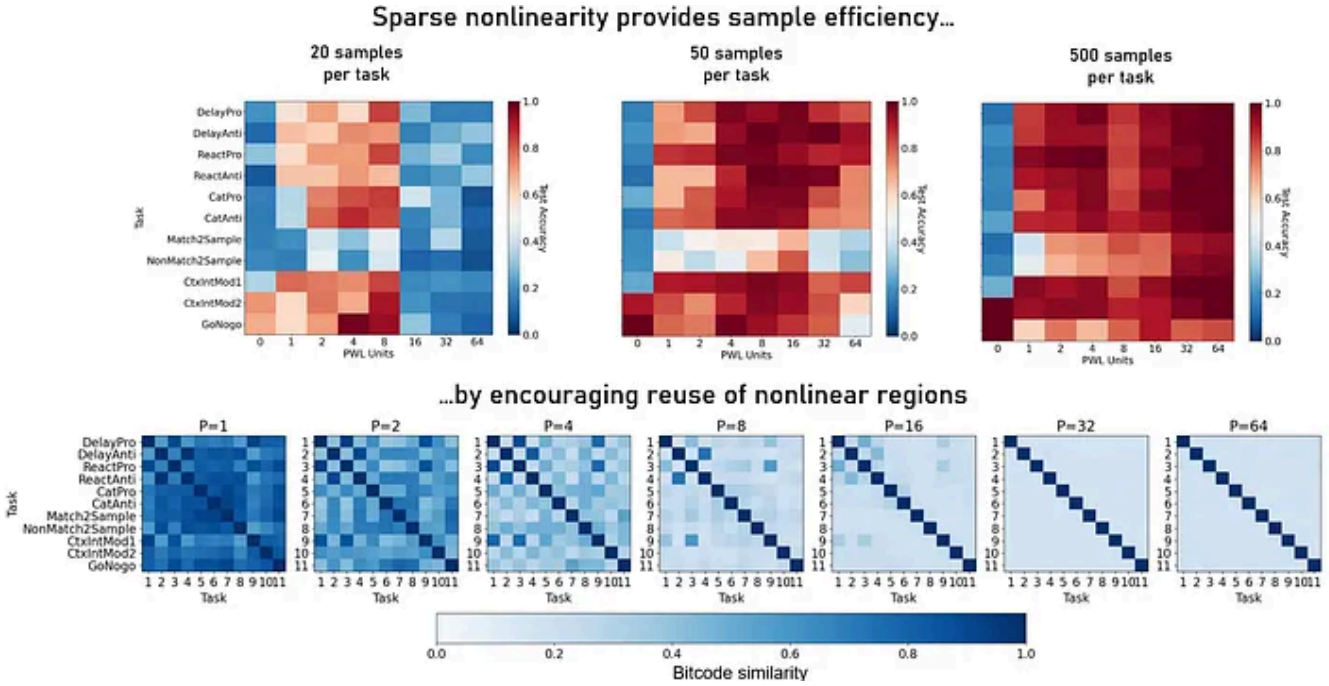
Encouraged by these results, we asked a broader question: how much nonlinearity is actually necessary across different sequence modeling tasks? Part of our motivation came from recent structured state space models like Hippo, S4, and Mamba. These architectures use linear recurrence because it enables the efficient parallelization typically associated with Transformers, while avoiding the quadratic scaling of attention mechanisms. Naively, this efficiency comes at the cost of expressivity, but it's not always clear whether that is actually a problem. Recent work has even suggested that **linear recurrent units can outperform nonlinear ones**, raising the question of whether recurrent nonlinearity might be not just unnecessary but actually harmful.

A challenge in shedding light on this question is that these models stack linear recurrence with nonlinear transformations across multiple layers, making it difficult to isolate what each component contributes, especially on simple tasks that should admit simple solutions. With the AL-RNN, we took a minimal approach by focusing on settings where recurrence takes center stage, and we can specifically investigate how changing nolinearity within the recurrence affects task solutions.

We tested AL-RNNs on a diverse benchmark suite spanning classic machine learning problems (sequence classification on text, images, and audio), algorithmic challenges (addition and copy tasks), cognitive tasks from computational neuroscience (contextual integration, stimulus selection, real neural recordings), and syntactic reasoning tasks that bridge toward language models. This breadth allowed us to distinguish which tasks require nonlinearity as a functional necessity versus those solvable by predominantly linear dynamics.

We found that constraining nonlinearity is not only useful for interpretability, but often actually leads to **improved performance**. On top, we find very clear task mechanisms depending on tasks. The most compelling evidence came from training on 11 cognitive tasks simultaneously, here using variants of delayed response, match-to-sample, and context integration tasks that share temporal structure but require distinct computations. With very limited data (20 samples per task), sparse nonlinearity (P=1 to 8 units) dramatically outperformed both linear and fully nonlinear models. At intermediate data levels (50 samples), sparse models maintained strong performance while fully nonlinear models began catching up. With abundant data (500 samples), fully nonlinear models

finally achieved the best performance, but sparse models remained competitive.



(Top) Test accuracy across 11 cognitive tasks as a function of nonlinearity level. (Bottom) Task similarity measured via Jensen-Shannon divergence between bitcode distributions reveals the mechanism.

Within the AL-RNN, the mechanism is transparent: sparse nonlinearity forces the network to reuse a small set of nonlinear computations across tasks. We quantified this by measuring which linear subregions each task occupied. With P=4, related tasks clustered together, e.g. pro-response variants shared subregions, while anti-response variants formed a separate group. This shared structure explained the sample efficiency advantage: the

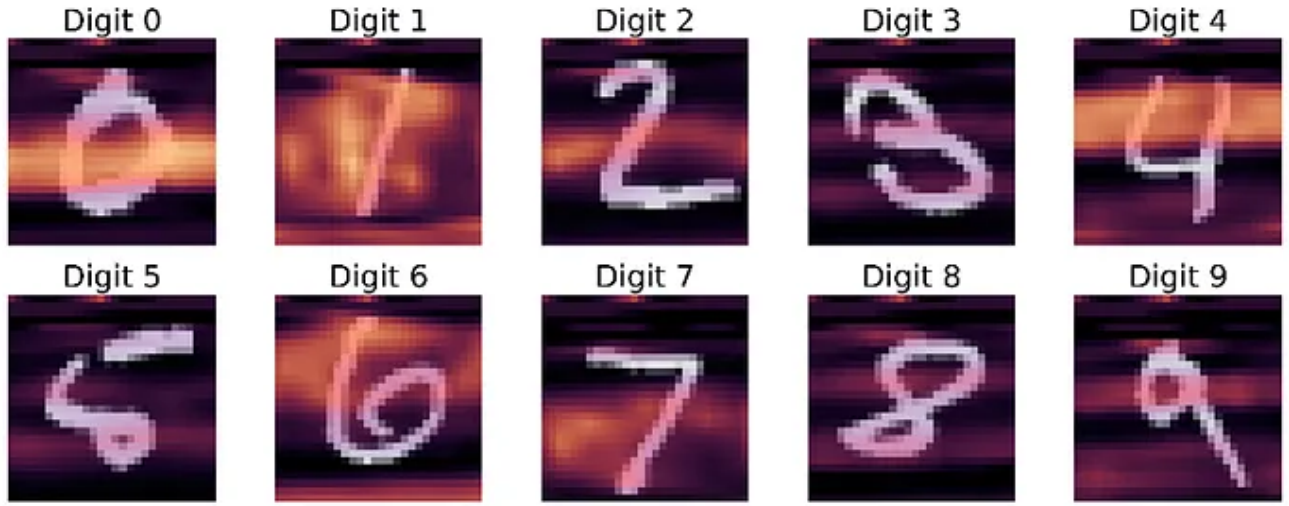
network learned cross-task generalizations when data was scarce. At full nonlinearity (P=64), this structure vanished as tasks spread across independent subregions without recurring patterns.

Beyond performance, the AL-RNN's piecewise-linear structure makes nonlinear computations explicit. By tracking which linear subregion is active over time (via "bitcodes"), we identified discrete computational primitives, such as gating, rule-based integration, context-dependent routing, and memory-dependent transients.

In the addition task, where the network must sum two marked inputs from a random stream, a single nonlinear unit sufficed, and the model learned an internal gating mechanism: linear units accumulated time through slow drift, while the nonlinear unit rapidly repositioned the state onto the appropriate integration trajectory when gated inputs occurred. This cleanly separated linear memory from the nonlinear task solution.

For classification tasks like sentiment analysis, image recognition, and speech commands, we found that sparse nonlinearity outperformed both fully linear and fully nonlinear models. The mechanism was clear from the

piecewise-linear structure. Each digit class aligned with its own set of linear subregions, enabling class-specific integration dynamics. All classes used slow integration (eigenvalues near 1), but subtle differences in these eigenvalues across subregions caused trajectories to diverge into well-separated clusters. Fully nonlinear models overdispersed this structure, fragmenting the computation across hundreds of subregions rather than maintaining a compact, interpretable solution.



In sequential MNIST, 28×28 digit images are flattened into sequences of 784 pixels, processed one at a time. The heatmaps overlay the average Hamming distance from a reference bitcode state, revealing when the network's nonlinear units switch between linear subregions. Switches align with visually informative stroke patterns: digits with sharp vertical onsets (1, 4, 6) trigger early activations, while digits with curved strokes (0, 8) activate later. Digits with similar overall shapes (8 vs. 9) show diverging bitcode patterns toward the end of the sequence.

In context-dependent tasks (both synthetic and from rodent neural recordings), single nonlinear units routed different task rules into distinct linear subregions, implementing the context-dependent switches that linear dynamics fundamentally cannot achieve. For compositional generalization in SCAN, the decoder's piecewise structure tiled latent space into syntax-aware subregions, with transitions between subregions synchronized to switches between linguistic constructs.

Across tasks, we consistently found that memory itself is often implemented via slow linear modes, **while computational operations are layered on top**

These findings reveal that sparse nonlinearity provides a **useful inductive bias** by encouraging networks to discover **simple, interpretable computational strategies** rather than distributing computation across many subregions and parameters. We think this is particularly exciting for applications where interpretability matters (which, arguably, should be most of them).

Within neuroscience, understanding task mechanisms underlying neural recordings has been a major focus for analyzing **how the brain implements computations across many different tasks in parallel**. This is crucial for understanding how the brain avoids catastrophic forgetting and enables transfer learning across tasks , something humans do effortlessly but AI systems still struggle with.

Our findings provide both the “what” and the “how.” We show how transfer learning emerges through computational reuse, and we do so in an interpretable backbone that’s dynamically transparent. This gives us two complementary lenses: eigenvalue analysis and stability theory from dynamical systems theory, plus a discrete view of computational motifs through switching between linear subregions.

While we focused on AL-RNNs for their analytical simplicity, the principle extends beyond this specific architecture. In the supplement, we experimented with restricting nonlinearity in more complex architectures like Transformers and S4. We found that placing nonlinearity strategically within the stack of layers, (e.g. only in the output layer rather than everywhere), improved performance on several tasks. The interpretability looks different there (these aren’t first-order Markov processes with analyzable dynamical systems properties), but the performance benefits

suggest that sparse nonlinearity may be a broadly useful design principle for modern sequence models.

Machine Learning | Artificial Intelligence | Large Language Models | Data Science | Science

Written by Manuel Brenner

4.2K followers · 85 following

Following

Research group leader AI for neuroscience and and psychiatry. Connect via LinkedIn: https://www.linkedin.com/in/manuel-brenner-772261191

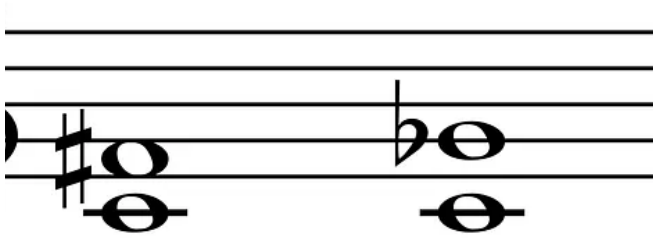
More from Manuel Brenner

Manuel Brenner

Confessions of a Bach Addict

"Music owes as much to Bach as religion to its founder." —Robert Schumann

Jun 16, 2019 42 5



Manuel Brenner

A physicist's perspective on tonality

Tonality is so deeply ingrained in the way we make and perceive music that it is very hard...

Nov 19, 2018 69 2

Manuel Brenner

Why the Poetry of the Tang Dynasty can still be relevant today

In TDS Archive by Manuel Brenner

Ants and the Problems with Neural Networks

After having re-read Guy Gavriel Kay's brilliant Under Heaven last week, I want to...

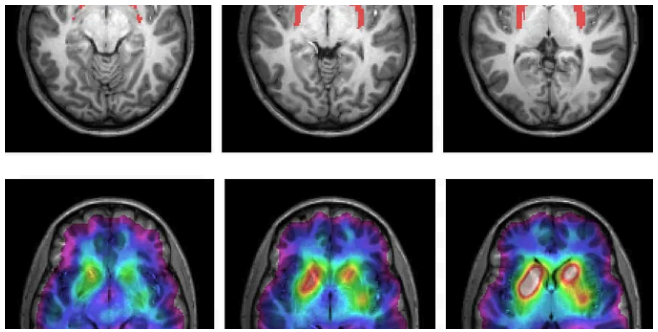
Sep 25, 2018 7

How Cognitive Science might Transform Neuroscience

Jul 6, 2019 603 2

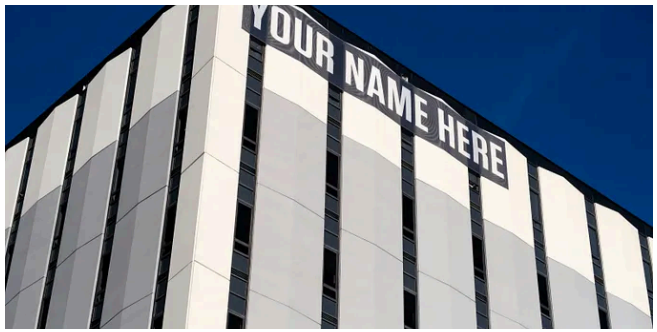
See all from Manuel Brenner

Recommended from Medium



In Write A Catalyst by Dr. Patricia Schmidt

As a Neuroscientist, I Quit These 5 Morning Habits That Destroy You...



DamenC

Fine-Tuning BERT for Named Entity Recognition: A Step-by-Step Guide

Most people do #1 within 10 minutes of waking (and it sabotages your entire day)

Jan 14 19.6K 331

Named Entity Recognition (NER) is a fundamental task in Natural Language...

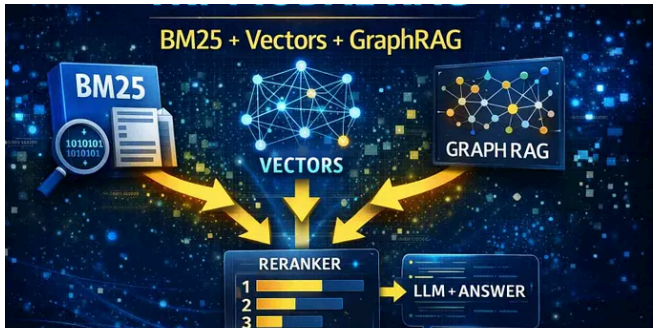
Oct 3, 2025 4

Rice Yang

How Transformers Become Faster And Smarter: From KV Cache to...

Explore the fundamental evolution of the LLM decoder inference

Aug 16, 2025 8

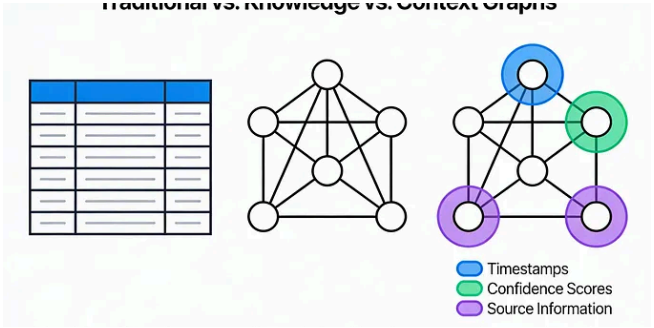


In Python in Plain English by Tarun Singh

Beyond Hybrid RAG That Actually Works: Vector + BM25 + GraphRA...

If you’re already using GraphRAG + Vector RAG, you’re ahead of most people.

Jan 18 30



In Neural Notions by Nikhil

What are Context Graphs: Building the AI that trulyUnderstands

Imagine an AI system that not only processes information but truly understands it ->...

Dec 24, 2025 33

Barack Obama

A Wake-Up Call for Every American

The killing of Alex Pretti is a heartbreaking tragedy. It should also be a wake-up call to...

2d ago 53K 726

See more recommendations