Self-Organizing Mixture of Experts (SOME): A Framework for In-Inference Adaptation by Decoupling Function from Address

Author: Focus Labs
Date: October 7, 2025

Abstract
The Mixture of Experts (MoE) paradigm has enabled neural networks to scale to trillions of parameters by routing inputs to a sparse subset of specialized subnetworks. However, these models are typically static post-deployment, lacking mechanisms for rapid adaptation to new data distributions without costly retraining. This paper introduces the Self-Organizing Mixture of Experts (SOME) architecture, a novel framework designed to enable continuous, in-inference adaptation while mitigating catastrophic forgetting. The core innovation of SOME is the strategic decoupling of an expert's static function (its frozen weights) from its dynamic address (a plastic routing key). This separation allows the model to reorganize the discoverability of its experts in response to real-time usage patterns. This adaptation is driven by a set of principled, gradient-free update rules consisting of two attractive forces—Query Pull (a centroid drift toward activating queries) and Peer Pull (a Hebbian co-activation coupling)—which are stabilized by countervailing mechanisms of Usage Inertia and Repulsive Decay.
This paper makes three primary contributions. First, we provide a formal definition of the SOME architecture and its dynamics, grounding its heuristic update mechanisms in a mathematical objective function. Second, we situate SOME within the broader academic landscape, clarifying its relationship to Self-Organizing Maps (SOMs) and differentiating it from established continual learning (CL) and test-time adaptation methods. Third, we present a comprehensive experimental plan to validate the architecture's core claims, providing a complete blueprint for developing and evaluating this novel approach to adaptive intelligence.

1. Introduction

A primary challenge in deploying large-scale neural networks is the tension between knowledge stability and adaptive plasticity. A model must retain its vast, pre-trained capabilities while simultaneously adapting to novel or shifting data distributions encountered during inference. Modifying a model's core weights often leads to catastrophic forgetting, where performance on previous tasks degrades rapidly. Conversely, a completely static model becomes brittle and loses performance as real-world data evolves. The Self-Organizing Mixture of Experts (SOME) architecture is proposed as a direct solution to this dilemma.
The central thesis of SOME is that a model can achieve in-inference adaptation by modifying how it accesses its knowledge rather than changing the knowledge itself. This is accomplished by separating an expert's immutable function (its frozen multi-million parameter weights) from its mutable address (a low-dimensional routing key). These keys exist in a high-dimensional vector space and are dynamically updated based on usage, effectively creating a self-organizing "address book" that continually refines itself.
This paper formally introduces the SOME architecture. We proceed as follows:

- Section 2 (Related Work): We ground SOME in the context of existing research in self-organizing systems, continual learning, and Mixture of Experts routing.

- Section 3 (Formal Methodology): We provide a complete technical definition of the architecture, its components, the mathematical basis for its update rules, and the crucial stability mechanisms.
- Section 4 (Experimental Validation Plan): We lay out a rigorous plan for empirical validation, including specific datasets, baselines, metrics, and ablation studies designed to test the core assumptions of the framework.
- Section 5 (Discussion): We analyze the system's potential, inherent limitations, and the technical challenges related to stability, scalability, and security.
- Section 6 (Conclusion): We conclude by summarizing the SOME framework and outlining a clear, actionable research agenda.

2. Related Work

The SOME architecture intersects with three key areas of machine learning research.

2.1. Self-Organizing Systems and Vector Quantization

The "Query Pull" mechanism, a core component of SOME, is functionally identical to the learning rule in Learning Vector Quantization (LVQ) and conceptually similar to the competitive learning process in Teuvo Kohonen's Self-Organizing Maps (SOMs). In these methods, prototype vectors (analogous to SOME's keys) are moved closer to the data points they represent. SOME builds upon this classic foundation but differentiates itself by:

1. Architectural Integration: SOME is not a standalone clustering algorithm but an integrated routing component within a deep neural network's MoE layer.
2. Compound Dynamics: The LVQ-style update is augmented by additional forces (Peer Pull, Inertia, Decay) designed to foster compositional structure and ensure long-term stability within the model.

2.2. Continual Learning (CL)

The problem of catastrophic forgetting is the central focus of Continual Learning. Mainstream CL paradigms include regularization-based methods (e.g., Elastic Weight Consolidation, or EWC), which penalize changes to important weights; replay-based methods (e.g., Gradient Episodic Memory, or GEM), which store and rehearse old data; and architectural methods that add new parameters for new tasks. SOME proposes an orthogonal approach: it performs no gradient-based updates on expert weights and adds no new network capacity. A key goal of future work will be to benchmark SOME against these established methods on standard CL tasks.

2.3. Mixture of Experts (MoE) Routing

Modern MoE architectures utilize sophisticated mechanisms to ensure load balancing and prevent "router collapse," where a few experts dominate. These often involve auxiliary loss functions that encourage uniform expert selection. A robust implementation of SOME must incorporate these lessons. The proposed "Usage Inertia" mechanism is a step in this direction, but production-grade systems will require explicit load-balancing controls to prevent expert "hotspotting" and "starvation."

3. Formal Methodology and Architecture

This section provides a formal, replicable definition of the SOME architecture.

3.1. Core Components

A SOME layer is comprised of static and dynamic components.

- Static Components (The Knowledge Base):
    - Expert Pool (E): A collection of M independent Feed-Forward Networks (FFNs), E = {e_1, e_2, ..., e_M}.
    - Expert Weights (W_up, W_down): The projection weight matrices for each expert are frozen after an initial pre-training phase.
- Dynamic Components (The Routing System):
    - Router (R): A small, trainable network that generates a query vector q from an input token's hidden state x.
    - Key Store (K): A dynamically updatable store containing a d_key-dimensional routing key k_i for each expert e_i. These keys are plastic and evolve during inference.

3.2. The Forward Pass

1. Query Generation: The router computes the query q = Q(x).
2. Expert Scoring: Similarity scores are computed between q and all expert keys k_i, typically via dot-product: s_i = q · k_i.
3. Top-K Selection: The K experts with the highest scores are selected for activation.
4. Gating and Combination: A softmax function is applied to the scores of the selected experts to determine their gating weights g_i. The final layer output is the weighted sum of the expert outputs, added to the input via a residual connection: y_output = x + Σ(g_i * e_i(x)).

3.3. The Dynamic Update Mechanism

After a batch of data is processed, the keys are updated in a gradient-free "consolidation" phase. These updates can be seen as approximating an optimization of a global objective function that encourages keys to be close to their assigned queries while maintaining structural stability.

- Force 1: Query Pull (Relevance Attraction)
    - Objective: To move an expert's key toward the centroid of the queries that activate it.
    - Update Rule: For each token-expert pair (q, e_i) in the Top-K set, the expert's key is updated: k_i_new = k_i_old + α * (q - k_i_old), where α is a small plasticity rate.
- Force 2: Peer Pull (Hebbian Co-activation)
    - Objective: To create conceptual bonds between experts that are frequently used together on the same input.
    - Update Rule: For each pair of experts (e_i, e_j) activated by the same query q, their keys are pulled closer symmetrically:
      k_i_new = k_i_old + β * (k_j_old - k_i_old)
      k_j_new = k_j_old + β * (k_i_old - k_j_old)
      where β is the bonding rate, typically β < α.

3.4. Stability Mechanisms

A system governed solely by attractive forces is inherently unstable. The following countervailing forces are critical for long-term equilibrium.

- Mechanism 1: Gravitational Mass (Usage Inertia)
    - Objective: To ensure that frequently used, generalist experts have more stable keys, anchoring the key space.
    - Implementation: The learning rates α and β are scaled by an expert's activation frequency, usage(e_i).
      α_effective = α / (1 + usage(e_i))
      β_effective = β / (1 + usage(e_i))
      The usage(e_i) term is calculated as an exponential moving average of an expert's activation over time.
- Mechanism 2: Dark Energy (Repulsive Decay)
    - Objective: To prevent the collapse of all keys into a single point and to prune unused experts.
    - Implementation: We employ a Decay to Origin mechanism for its computational efficiency (O(M)) and stability. Keys of experts whose usage falls below a threshold θ are decayed toward the origin vector.
      if usage(e_i) < θ: k_i_new = k_i_old * (1 - δ)
      where δ is a small, positive decay rate. This serves as a "forgetting" mechanism that removes irrelevant experts from the active pool over time.

4. Experimental Validation Plan

A rigorous experimental protocol is required to validate the claims of the SOME architecture.

4.1. Core Hypotheses

1. Stable Equilibrium: The interplay of the defined forces leads to a stable, non-collapsing, and semantically organized key space.
2. Forgetting Mitigation: Address-only updates significantly reduce catastrophic forgetting on sequential tasks compared to fine-tuning a static MoE model.
3. Emergent Composition: The Peer Pull mechanism creates functional "meta-experts" by co-locating the keys of complementary specialists.

4.2. Datasets and Baselines

- Datasets:
    1. Continual Learning Benchmarks: Standard datasets like Split CIFAR-100 and Permuted MNIST will be used to quantitatively measure forgetting.
    2. Domain Shift Benchmarks: A large text corpus partitioned by topic (e.g., technical, legal, fiction) will be used to test adaptation to distribution shifts.
- Baselines for Comparison:
    1. Static MoE (Frozen): An MoE model with no updates post-training.
    2. Static MoE (Fine-tuned): An MoE model where all weights are fine-tuned on new tasks.
    3. Standard CL Method: A representative continual learning baseline, such as Elastic Weight Consolidation (EWC).

4.3. Analysis and Metrics

- Quantitative Metrics: To measure performance on CL benchmarks, we will use standard metrics including Average Accuracy, Backward Transfer (BWT) to measure forgetting, and Forward Transfer (FWT) to measure adaptation to new tasks.

- Qualitative and Diagnostic Analysis:
    - Key Space Visualization: We will use dimensionality reduction techniques like t-SNE and UMAP to visualize the expert key space over time. This will allow for inspection of cluster formation, fragmentation, or collapse.
    - Expert Utilization Analysis: We will track the entropy of expert selection and the Gini coefficient of expert loads to monitor load balancing and prevent expert starvation.
- Ablation Studies: We will systematically disable each of the core mechanisms (Peer Pull, Inertia, Decay) to empirically demonstrate their contribution to the system's overall performance and stability.

5. Discussion of Limitations and Risks

The SOME framework, while promising, rests on several key assumptions and presents significant technical challenges.

- Semantic Organization: The framework assumes that the vector-arithmetic updates will organize the key space in a semantically meaningful way. This is not guaranteed and must be verified empirically.
- Router-Key Co-adaptation: The router is trained on an initial key distribution. As keys drift, the router's outputs may become mis-calibrated. Future work must investigate mechanisms for managing this, such as periodic key normalization or introducing a learnable temperature in the gating function.
- Systems Complexity: While conceptually simple, implementing a dynamic key store that can be updated at inference speed without introducing significant latency is a non-trivial systems engineering challenge, especially for models with millions of experts. Libraries like FAISS (Facebook AI Similarity Search) offer a potential path, but integration requires careful design.
- Security and Robustness: The key update mechanism is a potential attack surface. Adversarially crafted inputs could be used to "poison" the key space, degrading model performance. A production-ready system would require robust defenses, such as rate-limiting updates, employing outlier rejection, and implementing rollback mechanisms.

6. Conclusion

This paper has introduced the Self-Organizing Mixture of Experts (SOME), a novel architecture designed to enable adaptive intelligence in large-scale neural networks. By decoupling an expert's function from its address, SOME offers a path to in-inference learning that mitigates the risk of catastrophic forgetting. We have provided a formal definition of the architecture, grounded its dynamics in mathematical principles, and outlined a comprehensive research plan to validate its claims.

The path forward is clear: rigorous implementation and empirical validation are the essential next steps. The framework presented here provides a complete blueprint for this endeavor. If successful, the core principles of SOME could represent a significant step toward developing more autonomous, efficient, and continually learning AI systems.