Learning Like an Octopus: Octonet

A Decoupled Neural Architecture for Mitigating Catastrophic Forgetting via Selective Causal Imprinting

Ryuku Akahoshi

January 2, 2026

---

1. Introduction

1.1 The Structural Limits of Modern Deep Learning

Deep learning, particularly the rise of Transformer architectures, has revolutionized natural language processing and computer vision. Yet we still face two fundamental "physical and mathematical constraints."

First: **Catastrophic Forgetting**. In traditional models, all parameters are updated simultaneously based on a single loss function. This means acquiring new knowledge irreversibly destroys existing weight representations.

Second: **Computational Scalability**. Self-Attention mechanisms require $O(N^2)$ computation for input sequence length N, and retaining long-term memory within context hits the physical limits of VRAM.

1.2 Biological Inspiration: The Octopus as Distributed Intelligence

To solve these challenges, we looked to the octopus. About two-thirds of an octopus's neurons are distributed across its eight tentacles, with each arm capable of independent sensory processing and motor control. This distributed architecture doesn't require the central brain to handle every computational detail. Instead, peripheral units function as "local intelligence," exchanging only highly abstracted signals with the center.

1.3 Our Proposal: Octonet

We propose "Octonet," a radically new neural architecture mimicking this distributed structure. Octonet consists of a single **Head** for central control and a massive number of autonomous **Tentacles**.

The key innovation: we completely separate learning signals into "Announcement" and "Result." This allows us to imprint causal relationships onto specific units only when needed, achieving complete control over knowledge preservation and updates.

---

2. Model Architecture

Octonet's architecture spatially and temporally decouples information "processing" from "storage."

2.1 Signal Protocol Separation: Announcement and Result

Unlike traditional neural networks that always perform "prediction and error correction (BP) as a set" for input signals, Octonet splits signals into two phases:

Announcement Vector: A preparation signal sent from Head to Tentacles. This is abstract contextual information about "what kind of event or data is about to arrive." Upon receiving this signal, Tentacles make predictions based on their weights.

Result Vector: Confirmed information about what actually happened, or the ground truth data to be learned. Crucially, backpropagation (BP) only occurs in Tentacles when this result signal is sent. This allows Tentacles to switch between "knowledge retrieval (inference)" and "knowledge update (learning)" based on the Head's decision.

2.2 Tentacles: Autonomous Causal Modules

Tentacles have simple neural network (NN) structures. Their role is to maintain "causal structures" that map specific announcement vectors to appropriate result vectors.

Permanent Causal Storage: If the Head doesn't send a "result" signal, the weights inside a Tentacle never change. This means existing knowledge won't fade even after tens of thousands of inferences, structurally eliminating catastrophic forgetting.

Storage Efficiency: Since Tentacles are independent parameter sets, not all units need to reside in GPU memory. Inactive Tentacles can be offloaded to external storage like SSDs and loaded on-demand based on the Head's "announcements." This dramatically expands beyond traditional model size constraints.

2.3 Head: A Transformer-Based Meta-Learner

The Head adopts the most complex Transformer-type architecture and functions as the system's orchestrator.

Information Abstraction: The Head integrates environmental information with prediction results from Tentacles called by announcements. Since memory itself is handled by Tentacles, the Head's input sequence length depends only on "the number of Tentacles needed at that moment," eliminating the need to recursively process massive past contexts.

Active Thinking: The Head can execute "internal thought processes" like sending an announcement to a specific Tentacle and having another Tentacle judge whether that result is correct. This isn't passive information processing—it's an active process of exploring causal relationships through vector space operations.

---

3. Learning Mechanism & Meta-Cognition

Octonet's learning differs from traditional processes that minimize a single error function. Instead, it's defined as "dual-loop learning" where independent learning loops in the center (Head) and periphery (Tentacles) interact.

3.1 Hierarchical Learning Separation

Learning in Octonet is separated into two layers based on role:

Tentacle Local Learning (Supervised Learning): Each Tentacle performs extremely simple supervised learning, treating the "Announcement" from the Head as input x and the "Result" as label y. The goal is to acquire a function $f(x) \approx y$ that accurately approximates what should happen (result) from a specific context (announcement).

Head Meta-Learning (Reinforcement Learning-style Optimization): The Head learns a "policy" determining which Tentacle to send announcements to and when, and which Tentacles should receive results for learning (BP). This is meta-learning aimed at achieving the system's overall objectives (reward maximization), with the Head training each Tentacle as an "available tool."

3.2 Selective Backpropagation and Causal Imprinting

The core of this architecture is that the Head has complete authority over backpropagation (BP) execution.

Voluntary Updates: Result signals are sent and BP is executed only when the Head decides "I want this Tentacle to retain this knowledge." This prevents low-importance information or noise from destroying existing weights.

Delayed Causal Learning: We can place arbitrary time intervals between sending announcement signals and result signals. This allows us to store and learn "causal relationships between distant events"—difficult for traditional AI—as single training data pairs within specific Tentacles.

3.3 Self-Diagnosis and Failure Analysis

Octonet can retrospectively analyze why its reasoning failed by reproducing past causal structures.

Simulation via Re-announcement: When inference fails (reality diverges from prediction), the Head resends the "announcement" from the failure moment to the relevant Tentacle. Without sending a result signal, the Tentacle reproduces its prediction value without changing its weights.

Error Localization: By comparing reproduced values from each Tentacle with actual results, the Head can identify "which Tentacle's causal model was wrong." This process allows correcting (or reinitializing) only specific Tentacles with incorrect knowledge, without retraining the entire system.

---

4. Experimental Design & Theoretical Analysis

This section presents experimental scenarios to validate Octonet's effectiveness and analyzes its theoretical advantages over traditional Transformer models.

4.1 Experimental Scenario: Causal Inference from Local Information

Consider a task of partially observing a cat image through a "window" and identifying what it is.

Procedure: The Head sends actions like "move" or "acquire visual information" as announcement vectors to each Tentacle. By performing BP with the result "cat" on a specific Tentacle, we imprint the causal relationship between that pixel arrangement and the concept "cat" into that Tentacle.

Evaluation Metric: Measure the retention rate of recognition accuracy for the previously learned "cat" when continuously performing similar tasks with different animals (dogs, birds, etc.). Since Octonet can physically block BP, the retention rate should theoretically be 100%.

4.2 Computational Complexity Comparison

Comparing computational costs between traditional Transformer architecture and Octonet:

Transformer Complexity: For input sequence length N, Self-Attention computation is $O(N^2)$.

Octonet Complexity: Let M be the total number of Tentacles and K be the number simultaneously activated by the Head ($K \ll M$). The computational load the Head must process is $O(K^2)$, remaining constant regardless of total knowledge M.

Memory Efficiency: Memory is distributed across Tentacles stored in external storage like SSDs, so VRAM consumption is limited to the active Head and a few Tentacles.

4.3 Theoretical Defense Against Catastrophic Forgetting

In traditional deep learning, the weight update formula is θ_{t+1} = θ_t - η∇L. Since all knowledge shares the same θ, interference is unavoidable.

In Octonet, each Tentacle i's weights θ^(i) are controlled by a switch function S^(i) ∈ {0,1} (presence/absence of result signal) generated by the Head:

$$\theta_{t+1}^{(i)} = \theta_t^{(i)} - S_t^{(i)} \cdot \eta \nabla L^{(i)}$$

By setting S_t^(i) = 0, we can perform inference while completely freezing specific knowledge bases, mathematically suppressing catastrophic forgetting to zero.

---

5. Discussion & Future Work

The proposed Octonet presents a powerful solution to structural challenges in traditional monolithic neural networks, but several scientific and technical issues remain for practical implementation.

5.1 Meta-Learning Complexity in the Head Unit

The Head is both the "source of intelligence" and the greatest challenge in this model.

Routing Difficulty: Learning a meta-strategy to identify which individuals among massive Tentacles return optimal responses to current "announcements" is extremely difficult. Initially, the Head doesn't know which Tentacles hold which causalities, requiring reinforcement learning-style exploration of a vast search space.

Reward Design Challenge: In complex thought processes, the "credit assignment problem"—determining which Tentacle outputs contributed to final success—affects Head learning efficiency. This requires mechanisms for self-referencing past action logs using Tentacles that maintain history.

5.2 Signal Vector Consistency and Semantic Drift

Signals exchanged between Head and Tentacles are abstract vectors.

Meaning Space Degradation: As learning progresses and the Head's internal representations (meaning space) update, there's a risk that the meaning of "announcement" vectors previously written to other Tentacles changes relatively (semantic drift).

Toward Solutions: We anticipate needing protocols to partially fix signal vector bases or fine-tune (align) Tentacles in response to Head updates.

5.3 Scalability Using Storage Hierarchy

A notable feature of Octonet is that Tentacles can be stored in external memory like SSDs.

On-Demand Loading: Not needing to deploy all knowledge in VRAM—loading only Tentacles related to "announcements" the Head deems necessary into main memory—breaks the traditional LLM constraint of "model size = knowledge capacity."

Future Challenges: Minimizing latency from external memory loading through predictive prefetch algorithms and high-speed bus utilization will be key for hardware implementation.

---

6. Conclusion

This paper proposes "Octonet," a new AI architecture inspired by the octopus's distributed nervous system.

6.1 Research Summary

Octonet adopts an innovative protocol separating signals into "announcement" and "result," with the Head selectively controlling backpropagation (BP). This achieves three things:

1. **Non-Destructive Learning**: Protecting existing weights unless "result" signals are sent mathematically avoids catastrophic forgetting.
2. **Dramatic Computational Efficiency**: Offloading memory to external Tentacles solves the Head's input sequence length problem, keeping inference costs constant even as knowledge grows.
3. **Long-Range Causality Acquisition**: Linking temporally distant announcements and results enables imprinting deeper causal structures.

6.2 The Future of Intelligence

Octonet promotes a shift from current AI as "a single giant brain" toward "social or modularized brains." This structure—where the Head freely trains and utilizes Tentacles as "tools" or "memories"—computationally recreates the benefits of distributed intelligence that biological intelligence acquired through evolution.

This model offers a path toward next-generation artificial general intelligence (AGI) capable of true "lifelong learning" with superior resource efficiency.