Structured Knowledge Accumulation: An Autonomous Framework for Layer-Wise Entropy Reduction in Neural Learning

Bouarfa Mahi
Quantiota
Email: info@quantiota.org

February 20, 2025
Version 1.4

Abstract

We introduce the Structured Knowledge Accumulation (SKA) framework, which redefines entropy as a dynamic, layer-wise measure of knowledge alignment in neural networks: $H^{(l)} = -\frac{1}{\ln 2}\sum_{k=1}^{K} \mathbf{z}_k^{(l)} \cdot \Delta\mathbf{D}_k^{(l)}$, where $\mathbf{z}_k^{(l)}$ is the knowledge vector and $\Delta\mathbf{D}_k^{(l)}$ is the decision probability shift vector at layer l over K forward steps. Rooted in the continuous entropy formulation $H = -\frac{1}{\ln 2}\int z\,dD$, SKA derives the sigmoid function, $D_i^{(l)} = \frac{1}{1+e^{-z_i^{(l)}}}$, as an emergent property of entropy minimization. This approach generalizes to fully connected networks without backpropagation, with each layer optimizing locally by aligning $\mathbf{z}_k^{(l)}$ with $\Delta\mathbf{D}_k^{(l)}$, guided by $\cos(\theta_k^{(l)})$. Total network entropy, $H = \sum_{l=1}^{L} H^{(l)}$, decreases hierarchically as knowledge structures evolve. Offering a scalable, biologically plausible alternative to gradient-based training, SKA bridges information theory and artificial intelligence, with potential applications in resource-constrained and parallel computing environments.

1 Introduction

Entropy, classically defined by Shannon [1] as $H = -\sum p_i \log_2 p_i$, quantifies uncertainty in a static, discrete probabilistic system. While foundational, this formulation falls short of capturing the dynamic, continuous structuring of knowledge in intelligent systems like neural networks. The sigmoid function, $\sigma(z) = \frac{1}{1+e^{-z}}$, a cornerstone of artificial intelligence (AI), has lacked a theoretical basis beyond empirical utility. Conventional training via backpropagation, which propagates errors backward through the network, is computationally intensive and biologically implausible, constraining scalability and real-world applicability.

This article presents the Structured Knowledge Accumulation (SKA) framework, reimagining entropy as a continuous process of knowledge accumulation. We propose:

1. Entropy as a dynamic measure, expressed in layers as $H^{(l)} = -\frac{1}{\ln 2}\sum_{k=1}^{K} \mathbf{z}_k^{(l)} \cdot \Delta\mathbf{D}_k^{(l)}$, approximating the continuous $H = -\frac{1}{\ln 2}\int z\,dD$,
2. Structured knowledge (z) and its continuous accumulation as the foundation of learning,
3. A forward-only, backpropagation-free learning rule driven by local entropy minimization.

We derive the sigmoid from continuous entropy reduction, extend it to fully connected networks, and demonstrate learning through local alignment of knowledge with decision dynamics. This framework offers a novel perspective for AI, enhancing optimization efficiency, interpretability, and biological plausibility, with implications for scalable and distributed neural systems.

2 Redefining Entropy in the SKA Framework

Entropy traditionally quantifies uncertainty in probabilistic systems, but its classical form is static and discrete, limiting its applicability to dynamic learning processes like those in neural networks. In the Structured Knowledge Accumulation (SKA) framework, we redefine entropy as a continuous, evolving measure that reflects knowledge alignment over time or processing steps. This section contrasts Shannon's discrete entropy with our continuous reformulation, enabling the use of continuous decision probabilities and supporting the derivation of the sigmoid function through entropy minimization.

2.1 Classical Shannon Entropy

For a binary system with decision probability D, Shannon's entropy is:

$$H = -D \log_2 D - (1 - D) \log_2(1 - D) \tag{1}$$

Its derivative with respect to D is:

$$\frac{dH}{dD} = \log_2 \left(\frac{1 - D}{D} \right) \tag{2}$$

This formulation assumes D is a fixed probability, typically associated with discrete outcomes (e.g., 0 or 1). While foundational, it does not capture the continuous evolution of knowledge in a learning system, where D may vary smoothly as the network processes inputs. To address this, we seek a continuous entropy measure that accommodates dynamic changes in D, aligning with the SKA's focus on knowledge accumulation.

2.2 Entropy as Knowledge Accumulation

In SKA, we redefine entropy for a single neuron as a continuous process:

$$H = -\frac{1}{\ln 2} \int z \, dD \tag{3}$$

Here, z represents the neuron's structured knowledge, and dD is an infinitesimal change in the decision probability, treated as a continuous variable over the range $[0, 1]$. The factor $-\frac{1}{\ln 2}$ ensures alignment with base-2 logarithms, consistent with Shannon's information units. Unlike the static snapshot of Equation 1, this integral captures how entropy accumulates as z drives changes in D, reflecting a dynamic learning process.

For a layer l with n_l neurons over K forward steps, we approximate this continuous form discretely:

$$H^{(l)} = -\frac{1}{\ln 2} \sum_{k=1}^{K} \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)} \tag{4}$$

where $\mathbf{z}_k^{(l)} = [z_1^{(l)}(k), \ldots, z_{n_l}^{(l)}(k)]^T$ is the knowledge vector, $\Delta \mathbf{D}_k^{(l)} = [\Delta D_1^{(l)}(k), \ldots, \Delta D_{n_l}^{(l)}(k)]^T$ is the vector of decision probability shifts, and the scalar product is:

$$\mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)} = \sum_{i=1}^{n_l} z_i^{(l)}(k) \Delta D_i^{(l)}(k) \tag{5}$$

The total network entropy sums over all layers:

$$H = \sum_{l=1}^{L} H^{(l)} \tag{6}$$

Equation 3 is the core theoretical construct, with Equation 4 as its practical discrete approximation. As $K \to \infty$ and $\Delta\mathbf{D}_k^{(l)}$ becomes infinitesimally small, Equation 4 approaches the continuous integral, enabling us to model D as a smooth function of z. This continuous perspective is essential for deriving the sigmoid using dynamics in later sections, while the discrete form supports implementation in neural architectures.

2.3 Accumulated Knowledge

Knowledge accumulates over steps:

$$z_k = z_0 + \sum_{f=1}^{k} \Delta z_f \tag{7}$$

In a layer, $\mathbf{z}_k^{(l)}$ evolves, reducing $H^{(l)}$ as it aligns with $\Delta\mathbf{D}_k^{(l)}$.

3 Deriving the Sigmoid Function

The SKA framework posits that the sigmoid function emerges naturally from continuous entropy minimization, linking structured knowledge to decision probabilities. This section demonstrates that when D follows $D = \frac{1}{1+e^{-z}}$, the SKA entropy H_{SKA} equals the classical Shannon entropy H_{Shannon}, differing by a constant (zero). By leveraging the continuous formulation from Section 2, we derive this equivalence, reinforcing the framework's theoretical grounding.

3.1 Key Definitions

3.1.1 Shannon Entropy (for binary decisions)

For a binary system with continuous decision probability D:

$$H_{\text{Shannon}} = -D \log_2 D - (1-D)\log_2(1-D) \tag{8}$$

3.1.2 SKA Entropy (layer-wise, for a single neuron)

The SKA entropy, defined continuously as in Section 2.2, is:

$$H_{\text{SKA}} = -\frac{1}{\ln 2}\int z\, dD, \tag{9}$$

where $z = -\ln\left(\frac{1-D}{D}\right)$ relates knowledge to D, consistent with $D = \frac{1}{1+e^{-z}}$ as shown in Section 3.1.

3.2 Equivalence Proof

Substituting $z = -\ln\left(\frac{1-D}{D}\right)$ (or equivalently, $z = \ln\left(\frac{D}{1-D}\right)$) into H_{SKA}:

$$H_{\text{SKA}} = -\frac{1}{\ln 2}\int \ln\left(\frac{D}{1-D}\right) dD. \tag{10}$$

Evaluate the integral with substitution $u = D$, $du = dD$:

$$\int \ln\left(\frac{D}{1-D}\right) dD = D\ln\left(\frac{D}{1-D}\right) + \ln(1-D). \tag{11}$$

Substituting back:

$$H_{\text{SKA}} = -\frac{1}{\ln 2}\left[D \ln\left(\frac{D}{1-D}\right) + \ln(1-D)\right]. \tag{12}$$

Rewrite $\ln\left(\frac{D}{1-D}\right) = \ln D - \ln(1-D)$:

$$H_{\text{SKA}} = -\frac{1}{\ln 2}\left[D \ln D - D\ln(1-D) + \ln(1-D)\right]. \tag{13}$$

Factorize:

$$H_{\text{SKA}} = -\frac{1}{\ln 2}\left[D \ln D + (1-D)\ln(1-D)\right]. \tag{14}$$

Thus:

$$H_{\text{SKA}} = H_{\text{Shannon}}. \tag{15}$$

3.3 Implications

- **Zero Difference**: The SKA and Shannon entropies are identical (differing by zero) when $D = \frac{1}{1+e^{-z}}$, confirming the sigmoid as an emergent property of continuous entropy reduction.
- **Knowledge Alignment**: This equivalence stems from z structuring D to minimize uncertainty, as defined in Sections 2 and 3.

3.4 Significance

1. **Theoretical Consistency**: SKA extends Shannon entropy into a continuous, dynamic context while preserving its core properties for sigmoidal outputs.
2. **Backpropagation-Free Learning**: Since $H_{\text{SKA}} = H_{\text{Shannon}}$, layer-wise entropy minimization aligns with classical uncertainty reduction, achieved via forward dynamics alone.
3. **Biological Plausibility**: The continuous, local alignment of z with D mirrors plausible neural learning mechanisms.

3.5 Summary

When D is the sigmoid function, H_{SKA} matches H_{Shannon} exactly, with a difference of zero. This result, derived from the continuous entropy $H_{\text{SKA}} = -\frac{1}{\ln 2}\int z\,dD$, validates SKA's foundation and its seamless integration with information theory, leveraging continuous dynamics for neural learning with classical information theory.

4 The Fundamental Law of Entropy Reduction

The SKA framework establishes a fundamental law governing how entropy decreases as structured knowledge evolves. This section derives this law using continuous dynamics, reflecting the continuous nature of decision probabilities and knowledge accumulation introduced in Sections 2 and 3. We then provide a discrete approximation for practical implementation, ensuring the framework's applicability to neural networks while rooting it in a continuous theoretical foundation.

4.1 Continuous Dynamics

For a single neuron, the rate of entropy change with respect to structured knowledge z is derived from the continuous entropy $H = -\frac{1}{\ln 2}\int z\,dD$. Taking the partial derivative:

$$\frac{\partial H}{\partial z} = -\frac{1}{\ln 2} z D(1-D) \tag{16}$$

This follows from $D = \frac{1}{1+e^{-z}}$ (as derived in Section 4), where $\frac{dD}{dz} = D(1-D)$, and reflects the neuron's local contribution to entropy reduction. For a layer l with n_l neurons at step k, this extends to each neuron i:

$$\frac{\partial H^{(l)}}{\partial z_i^{(l)}(k)} = -\frac{1}{\ln 2} z_i^{(l)}(k) D_i^{(l)}(k)\left(1 - D_i^{(l)}(k)\right) \tag{17}$$

Equation 17 governs the continuous reduction of layer-wise entropy $H^{(l)}$, driven by the alignment of $z_i^{(l)}(k)$ with the sigmoidal decision probability $D_i^{(l)}(k)$. This dynamic, localized process underpins the SKA's forward-only learning mechanism, leveraging the continuous evolution of D over time or input processing.

4.2 Discrete Dynamics

In practice, neural networks operate over discrete forward steps. For a single neuron at step k, the entropy gradient approximates the continuous form, incorporating the change in decision probability $\Delta D_k = D_k - D_{k-1}$:

$$\frac{\partial H}{\partial z}\Big|_k = -\frac{1}{\ln 2}\left[z_k D_k(1 - D_k) + \Delta D_k\right] \tag{18}$$

For layer l at step k, this becomes:

$$\frac{\partial H^{(l)}}{\partial z_i^{(l)}(k)} = -\frac{1}{\ln 2} z_i^{(l)}(k)\left[D_i^{(l)}(k)\left(1 - D_i^{(l)}(k)\right) + \Delta D_i^{(l)}(k)\right] \tag{19}$$

Equation 19 adapts the continuous law to discrete steps, where $\Delta D_i^{(l)}(k)$ captures the step-wise shift in $D_i^{(l)}(k)$. While Equation 17 represents the ideal continuous dynamics, Equation 19 provides a computable approximation, aligning knowledge adjustments with observed changes in decision probabilities across discrete iterations.

5 Generalization to Fully Connected Networks

The SKA framework extends seamlessly from single neurons to fully connected neural networks, leveraging the continuous entropy reduction principles established earlier. For a network with L layers, knowledge and decision probabilities evolve hierarchically, reducing total entropy through local, forward-only adjustments. This section outlines how SKA operates across layers, maintaining its biologically plausible and scalable design.

For a network with L layers:

- $\mathbf{z}_k^{(l)} = \mathbf{W}^{(l)}\mathbf{x}_k^{(l-1)} + \mathbf{b}^{(l)}$, the knowledge vector at layer l and step k,
- $\mathbf{D}_k^{(l)} = \sigma(\mathbf{z}_k^{(l)})$, the decision probabilities derived via the sigmoid function,
- $\Delta\mathbf{D}_k^{(l)} = \mathbf{D}_k^{(l)} - \mathbf{D}_{k-1}^{(l)}$, the step-wise shift in decision probabilities.

Layer-wise entropy, rooted in the continuous formulation, is approximated discretely:

$$H^{(l)} = -\frac{1}{\ln 2} \sum_{k=1}^{K} \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)} \tag{20}$$

The alignment between knowledge and decision shifts is quantified as:

$$\mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)} = \|\mathbf{z}_k^{(l)}\| \|\Delta \mathbf{D}_k^{(l)}\| \cos(\theta_k^{(l)}) \tag{21}$$

Total network entropy aggregates across layers:

$$H = \sum_{l=1}^{L} H^{(l)} \tag{22}$$

Learning proceeds by aligning $\mathbf{z}_k^{(l)}$ with $\Delta \mathbf{D}_k^{(l)}$ at each layer, reducing $H^{(l)}$ locally without requiring backward error propagation. In the continuous limit, this alignment reflects a smooth evolution of knowledge, approximated here by discrete steps for computational feasibility.

6 Learning Without Backpropagation

SKA achieves learning through localized entropy minimization, eliminating the need for backpropagation by leveraging forward-only dynamics. This section details the weight update mechanism and supporting metrics, grounded in the continuous entropy reduction law, and adapted for discrete implementation in fully connected networks.

Entropy minimization at layer l is driven by:

$$\frac{\partial H^{(l)}}{\partial w_{ij}^{(l)}} = -\frac{1}{\ln 2} \sum_{k=1}^{K} \frac{\partial (\mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)})}{\partial w_{ij}^{(l)}} \tag{23}$$

The update rule adjusts weights forward:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial H^{(l)}}{\partial w_{ij}^{(l)}} \tag{24}$$

Here, $\Delta D_i^{(l)}(k)$ is computed directly from forward passes, bypassing the need for error backpropagation. This local adjustment aligns with the continuous dynamics of knowledge evolution, approximated over discrete steps.

Step-wise Entropy Change

To quantify knowledge accumulation, the step-wise entropy change at layer l and step k is:

$$\Delta H_k^{(l)} = H_k^{(l)} - H_{k-1}^{(l)} = -\frac{1}{\ln 2} \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)} \tag{25}$$

This measures uncertainty reduction as $\mathbf{z}_k^{(l)}$ aligns with $\Delta \mathbf{D}_k^{(l)}$, with total layer entropy as:

$$H^{(l)} = \sum_{k=1}^{K} \Delta H_k^{(l)} \tag{26}$$

Entropy Gradient

The gradient of $H^{(l)}$ with respect to $\mathbf{z}_k^{(l)}$ at step k is:

$$\nabla H^{(l)} = \frac{\partial H^{(l)}}{\partial \mathbf{z}_k^{(l)}} = -\frac{1}{\ln 2}\mathbf{z}_k^{(l)} \odot \mathbf{D}_k'^{(l)} + \Delta \mathbf{D}_k^{(l)} \tag{27}$$

where $\mathbf{D}_k'^{(l)} = \mathbf{D}_k^{(l)} \odot (\mathbf{1} - \mathbf{D}_k^{(l)})$ is the sigmoid derivative. This gradient guides updates to minimize $H^{(l)}$, aligning knowledge with decision shifts.

Knowledge Evolution Across Layers

The gradient $\nabla H^{(l)} = -\frac{1}{\ln 2}\mathbf{z}_k^{(l)} \odot \mathbf{D}_k'^{(l)} + \Delta \mathbf{D}_k^{(l)}$ drives entropy reduction in layer l at step k. As $\mathbf{D}_k^{(l-1)}$ feeds into $\mathbf{z}_k^{(l)}$, each layer adapts uniquely—extracting broad features early and refining decisions later—mirroring a continuous knowledge flow approximated discretely.

Governing Equation of SKA

The network evolves according to:

$$\nabla H^{(l)} + \frac{1}{\ln 2}\mathbf{z}_k^{(l)} \odot \mathbf{D}_k'^{(l)} + \Delta \mathbf{D}_k^{(l)} = 0 \tag{28}$$

where $\nabla H^{(l)}$ minimizes entropy layer-wise, with updates following $-\nabla H^{(l)}$ to align $\mathbf{z}_k^{(l)}$ with $\Delta \mathbf{D}_k^{(l)}$.

Inter-Layer Entropy Change

The entropy change between layers l and $l+1$ at step k is:

$$\Delta H_k^{(l,l+1)} = -\frac{1}{\ln 2}\left[\mathbf{z}_k^{(l+1)} \cdot \Delta \mathbf{D}_k^{(l+1)} - \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)}\right] \tag{29}$$

This quantifies the spatial evolution of knowledge, complementing the temporal guidance of $\nabla H^{(l)}$, as entropy decreases through the network.

7 Application to Neural Networks

SKA structures knowledge hierarchically across layers, reducing total entropy H through continuous dynamics approximated over discrete steps. A multilayer perceptron (MLP) can train by minimizing H, with $\cos(\theta_k^{(l)})$ serving as a practical metric to monitor alignment between $\mathbf{z}_k^{(l)}$ and $\Delta \mathbf{D}_k^{(l)}$. This forward-only process leverages the framework's scalability and autonomy, applicable to diverse network architectures.

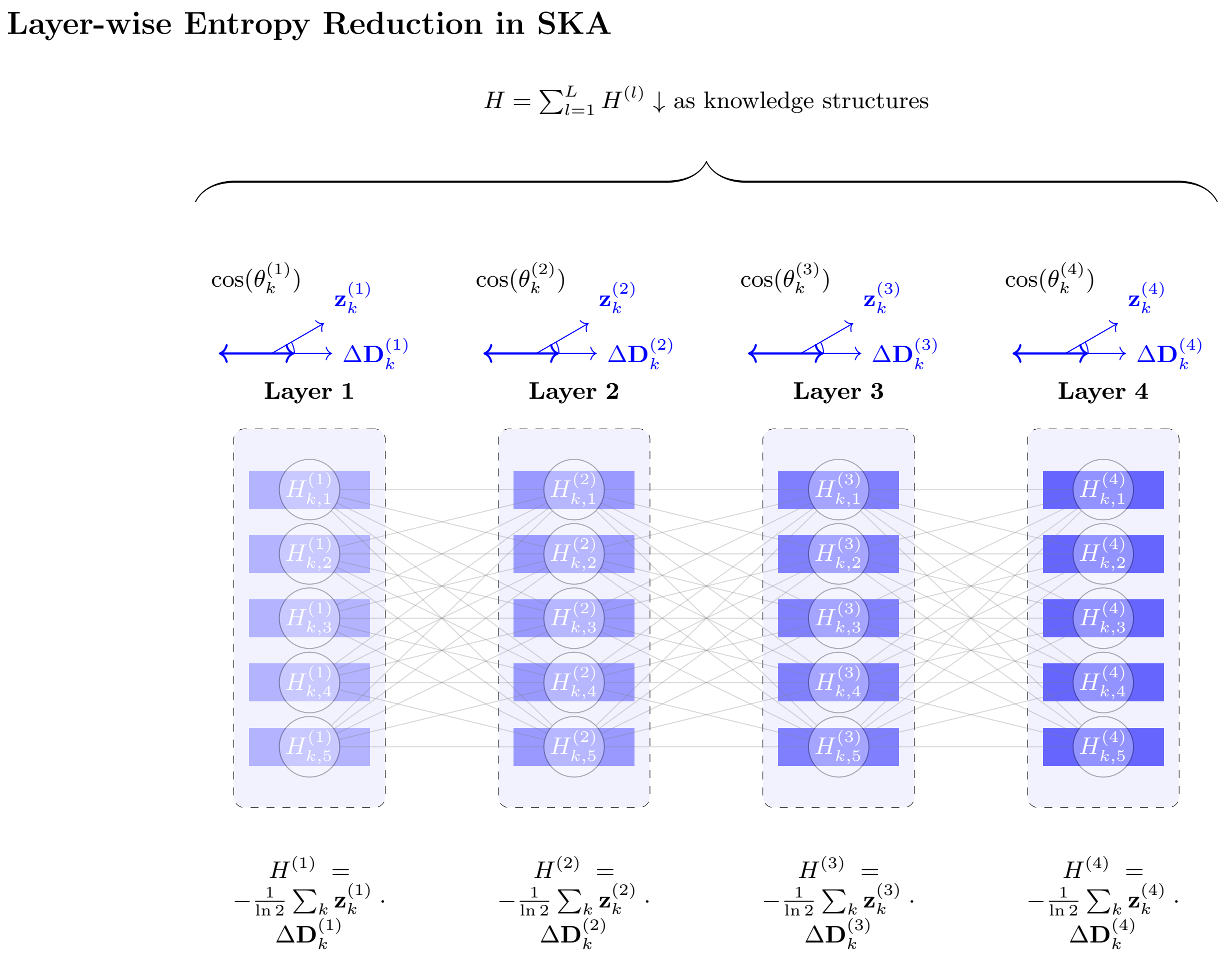

Figure 1: Layer-wise entropy reduction across layers at step k. Color intensity represents entropy level (darker blue = lower entropy), decreasing from Layer 1 to Layer 4. Each layer locally minimizes entropy by aligning knowledge vectors $\mathbf{z}_k^{(l)}$ with decision change vectors $\Delta\mathbf{D}_k^{(l)}$, measured by $\cos(\theta_k^{(l)})$.

8 Tensor-Based Implementation of SKA

To enhance computational efficiency and scalability, we introduce a **tensor-based implementation** of the Structured Knowledge Accumulation (SKA) framework. This approach preserves the theoretical foundations of SKA while leveraging tensor operations to enable efficient, parallelizable learning. By structuring SKA's mathematical formulation within a tensor framework, we streamline the **computation of knowledge accumulation, entropy reduction, and decision updates** across multiple layers and time steps.

Tensor Definitions

A neural network operating under SKA can be represented using the following tensors:

- **Knowledge Tensor** ($\mathbf{Z}$): Encodes structured knowledge for each neuron across **layers** (L), **steps** (K), and **neurons** ($n_{\max}$).
- **Decision Probability Tensor** ($\mathbf{D}$): Stores the neuron activations as sigmoid-transformed knowledge values.

- **Shift Tensor** ($\Delta\mathbf{D}$): Captures **local probability shifts** between consecutive steps, reflecting SKA's entropy-driven learning process.
- **Weight Tensor** ($\mathbf{W}$) and **Bias Tensor** ($\mathbf{b}$): Define learnable parameters that evolve dynamically to optimize knowledge accumulation.

Forward-Only Learning and Knowledge Update

SKA updates knowledge through a forward-only mechanism:

$$\mathbf{Z} = \mathbf{W} \cdot \mathbf{X} + \mathbf{b}$$

where $\mathbf{X}$ is the input tensor. Unlike backpropagation, SKA **does not require gradients to propagate backward**, making it more computationally efficient.

Entropy Computation and Learning

SKA's entropy formulation can be naturally expressed using tensor operations:

$$\mathbf{H} = -\frac{1}{\ln 2} \sum_{k=1}^{K-1} \mathbf{Z}_{k+1} \cdot \Delta\mathbf{D}_k$$

where **entropy decreases layer by layer** as structured knowledge accumulates.

To optimize learning, entropy gradients are computed as:

$$\nabla\mathbf{H} = -\frac{1}{\ln 2}\mathbf{Z} \odot \mathbf{D}' + \Delta\mathbf{D}$$

where $\mathbf{D}'$ is the derivative of the sigmoid function.

Weight Updates and Learning Stability

Weight updates follow a direct entropy minimization rule:

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial \mathbf{H}}{\partial \mathbf{W}}$$

Additionally, an **alignment tensor** monitors whether knowledge updates follow expected trajectories:

$$\Theta_{l,k} = \cos(\theta_{k+1}^{(l)}) = \frac{\mathbf{Z}_{l,k+1} \cdot \Delta\mathbf{D}_{l,k}}{\|\mathbf{Z}_{l,k+1}\| \|\Delta\mathbf{D}_{l,k}\|}$$

This ensures SKA maintains a **structured** and **controlled** learning process.

Advantages of the Tensor-Based Implementation

- **Parallel Computation**: Tensor operations allow efficient computation across multiple neurons and layers.
- **Scalability**: The framework extends naturally to large-scale networks without increasing complexity.
- **Biologically Inspired Learning**: Forward-only updates make SKA suitable for real-time, **low-energy computing environments**.

This tensor-based formulation further strengthens SKA's position as a **computationally efficient, biologically plausible, and scalable learning paradigm**. Future research will benchmark its performance against traditional backpropagation models, exploring its potential in edge computing, real-time AI, and neural network optimization.

9 Entropy Evolution in SKA

A key insight emerging from the SKA framework is the structured evolution of entropy across layers during learning. Unlike traditional deep networks, where entropy dynamics vary non-uniformly across layers, SKA exhibits a remarkable phenomenon: **all layers tend to converge toward a common entropy equilibrium value**. This behavior suggests that entropy is not merely decreasing but self-organizing in a structured manner.

9.1 Empirical Observation: Layer-Wise Entropy Convergence

Figure 2 presents the entropy evolution of different layers over multiple forward learning steps. The key observations are:

- **Layer-Specific Minima**: Each layer reaches a local minimum entropy value at different forward steps (K), before increasing slightly and stabilizing.
- **Convergence Toward a Common Equilibrium**: The entropy of layers 2, 3, and 4 asymptotically approaches a shared value around step $K = 49$, suggesting a fundamental equilibrium phenomenon.
- **Slow Convergence of Layer 1**: While layer 1 exhibits similar entropy reduction behavior, it approaches the equilibrium point more gradually, indicating a foundational role in hierarchical knowledge structuring.

This structured entropy evolution indicates that SKA does not merely reduce entropy but dynamically redistributes it across layers in a systematic fashion.

9.2 Theoretical Interpretation

The observed entropy equilibrium aligns with the SKA principle that **knowledge alignment drives learning**. The minimization of entropy at each layer leads to a balance where knowledge accumulation is harmonized across layers, resulting in a structured learning process.

This behavior suggests the existence of a fundamental law in SKA-based neural networks:

> *In an SKA neural network, layer-wise entropy evolves towards an equilibrium state, where knowledge accumulation stabilizes across hierarchical representations.*

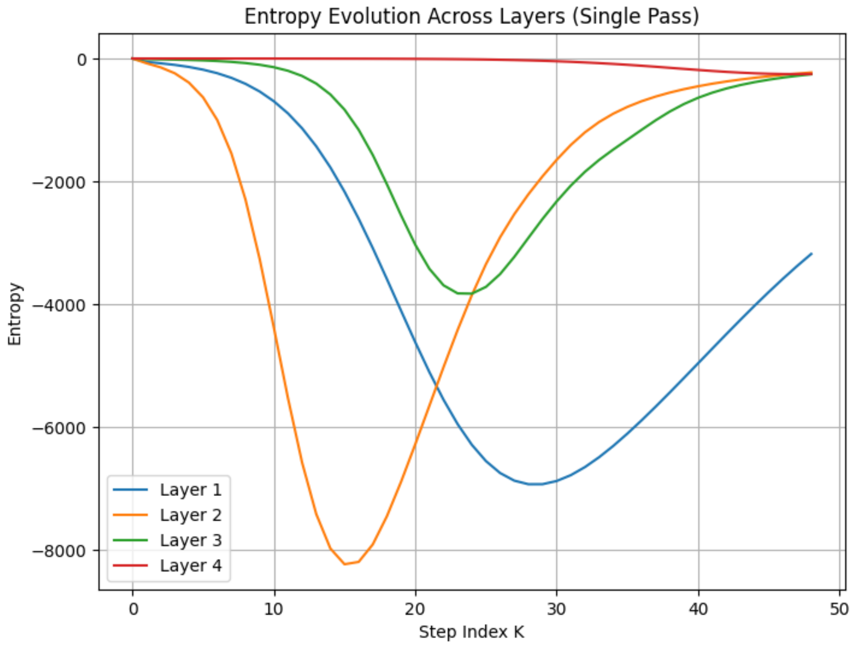

Figure 2: Entropy evolution across layers in an SKA neural network. Layers 2, 3, and 4 converge to a shared entropy equilibrium at $K = 49$, while Layer 1 approaches this point gradually.

9.3 Cosine Alignment Evolution

In addition to entropy evolution, SKA demonstrates another remarkable phenomenon: **cosine alignment evolution**. Figure 3 shows how the alignment between knowledge vectors $\mathbf{z}_k^{(l)}$ and decision probability shifts $\Delta\mathbf{D}_k^{(l)}$ evolves over time.

- **Cosine Alignment Convergence**: Across layers, $\cos(\theta_k^{(l)})$ converges toward a stable value, indicating increasing coherence between knowledge accumulation and decision updates.
- **Parallel Behavior to Entropy**: The stabilization of cosine values aligns with entropy reaching an equilibrium, reinforcing the structured learning dynamics of SKA.
- **Layer-Wise Synchronization**: The synchronization of alignment across layers suggests a self-organizing mechanism, optimizing knowledge updates through entropy minimization.

This empirical evidence further supports the hypothesis that SKA exhibits a natural, structured progression of learning, distinct from conventional gradient-based methods.

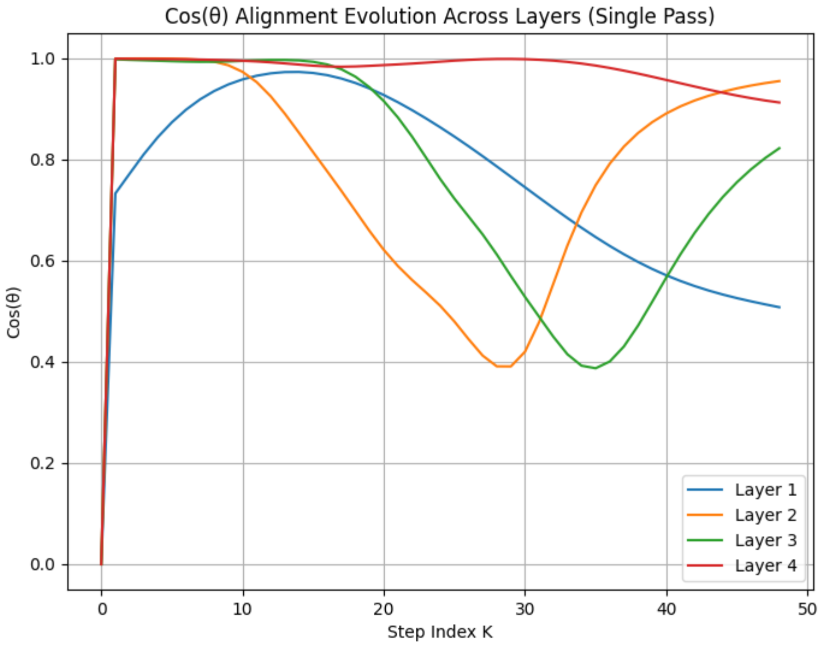

Figure 3: Cosine alignment evolution in SKA. The values of $\cos(\theta_k^{(l)})$ stabilize over forward steps, indicating structured knowledge alignment across layers.

The structured nature of entropy and cosine alignment evolution offers a novel perspective on learning in neural networks and may provide deeper insights into how biological and artificial learning systems self-organize their internal representations.

9.4 Output Decision Probability Evolution

Beyond entropy and cosine alignment, another critical visualization in SKA is the **evolution of output decision probabilities** over forward steps. Figure 4 illustrates the distribution of mean decision probabilities across all 10 classes as learning progresses.

- **Gradual Separation of Classes**: The decision probabilities shift over time, showing how SKA refines its class separability without explicit gradient updates.
- **Emergent Stability**: As the steps increase, decision probabilities stabilize, indicating convergence toward structured classification.
- **No Drastic Separation**: Unlike classical models, SKA does not create sharp decision boundaries but gradually refines knowledge accumulation to maintain a probabilistic structure.

This visualization provides further evidence of SKA's **self-organizing, entropy-driven classification process**, reinforcing its fundamental distinction from traditional backpropagation-based learning.

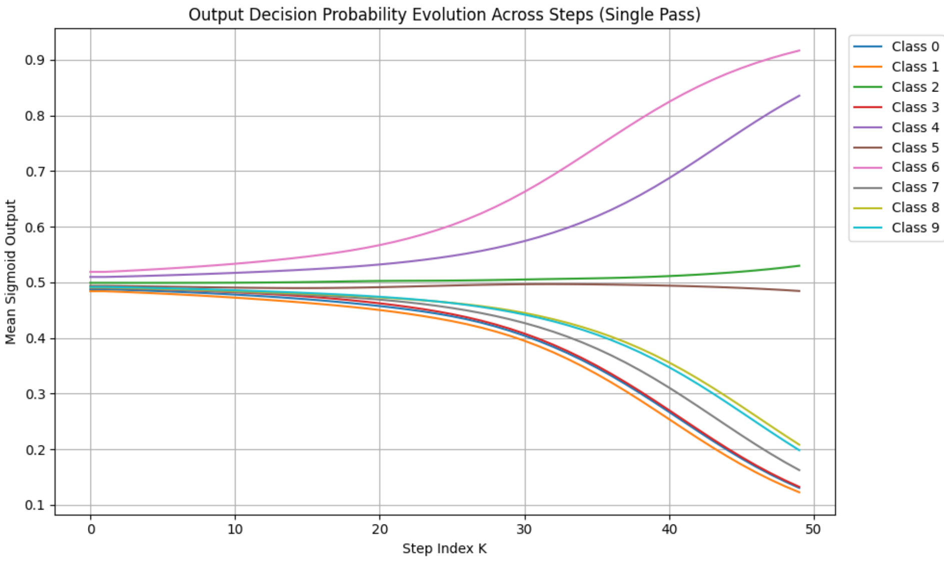

Figure 4: Evolution of output decision probabilities in SKA over forward steps. Unlike conventional models, SKA refines class separability progressively while maintaining structured probability distributions.

10 Implications and Future Work

The SKA framework offers transformative potential for neural network design and application, rooted in its continuous entropy reduction principles. Key implications and research directions include:

- **Autonomous Learning**: Layers independently minimize entropy, enabling self-sufficient training without global coordination.
- **Decentralization**: Local updates allow distribution across separate hardware, enhancing scalability in parallel systems.
- **Efficiency**: Forward-only learning reduces memory demands compared to backpropagation-based methods.
- **Interpretability**: Monitoring $\cos(\theta_k^{(l)})$ provides insight into layer-wise knowledge alignment.
- **Experiments**: Future work will compare SKA's performance to stochastic gradient descent (SGD) on datasets like MNIST, validating its efficacy.
- **Real-Time Processing**: Single-pass updates suit SKA for processing live data streams efficiently.

11 Conclusion

The SKA framework reimagines neural networks as systems that structure knowledge to minimize entropy, deriving the sigmoid function from continuous first principles and eliminating backpropagation. By leveraging local, forward-only dynamics, SKA offers a scalable, biologically inspired paradigm for AI, bridging information theory and neural learning with broad applicability.

References

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.