From Bayesian Inference to Neural Computation: The Analytical Emergence of Neural Network Structure from Probabilistic Relevance Estimation

Jaepil Jeong
Cognica, Inc.
Email: jaepil@cognica.io
Date: February 6, 2026

"The theory of probabilities is at bottom nothing but common sense reduced to calculus."
— Pierre-Simon Laplace, *Théorie analytique des probabilités*, 1812

Abstract

We demonstrate that the computational structure of a two-layer feedforward neural network with sigmoid activations emerges analytically from first-principles Bayesian inference over multiple relevance signals in information retrieval. Starting from a single question — *what is the probability that a document is relevant given multiple evidence signals?* — we apply Bayes' theorem to derive sigmoid calibration of individual scores, introduce a log-odds conjunction framework that resolves the well-known shrinkage problem of naive probabilistic conjunction, and show that the resulting end-to-end computation is formally isomorphic to a feedforward neural network: inputs pass through sigmoid activations (Bayesian calibration), undergo a linear transformation in log-space (geometric mean aggregation with additive bias), and pass through a second sigmoid activation (posterior computation). Crucially, this neural structure is not designed but *derived* — no architectural choices are made; the structure follows from algebraic necessity. We prove that the sigmoid's recurrence across both layers is a consequence of the exponential family structure of Bernoulli random variables, and independently derive the ReLU activation as the MAP estimator under sparse non-negative priors — establishing that the two dominant activations in deep learning answer complementary probabilistic questions: *"how probable?"* (sigmoid) and *"how much?"* (ReLU). We show that WAND and Block-Max WAND algorithms from information retrieval constitute exact neural pruning methods with formal safety guarantees — a property enabled by sigmoid's boundedness and unattainable with ReLU's unboundedness, both properties now understood as consequences of the respective probabilistic questions. Furthermore, relaxing the assumption of uniform signal reliability naturally extends the derived structure to the attention mechanism, identifying it as a form of context-dependent Bayesian model averaging and providing a probabilistic justification for why attention computes a weighted sum. We establish that the derived inference unit is the atomic building block of arbitrarily deep networks: depth arises from iterated marginalization over latent variables, where each layer constructs the evidence required by the next. Our results reverse the conventional direction of explanation in neural network theory: rather than building neural networks and analyzing them probabilistically, we begin with probability and arrive at neurons, activations, attention, and depth. The correspondence between activation functions and probabilistic questions reframes architecture design as question sequencing — choosing the order of probabilistic questions posed to the data — and enables a new form of interpretability in which the activation function of each layer identifies the type of inference it performs. This provides an existence proof that neural architectures can arise as theorems of probabilistic reasoning, with immediate implications for

network interpretability, efficient inference, and the theoretical foundations of neural computation.

---

1. Introduction

1.1 Background and Motivation

The relationship between probabilistic inference and neural computation has been a subject of sustained inquiry across multiple disciplines. The standard direction of investigation proceeds from neural networks to probabilistic interpretation: one constructs a neural architecture and subsequently asks what probabilistic model it corresponds to. Bayesian neural networks (Neal, 1996), variational inference methods (Blundell et al., 2015), and probabilistic deep learning (Gal & Ghahramani, 2016) all follow this direction.

In this paper, we demonstrate that the reverse direction is also productive — and yields a concrete, fully traceable result. We begin with a purely probabilistic question in information retrieval and show that the answer, when derived analytically, produces the computational structure of a feedforward neural network.

1.2 The Probabilistic Relevance Gap

Robertson (1977) introduced the Probability Ranking Principle (PRP), establishing that optimal document retrieval is achieved by ranking documents in decreasing order of their probability of relevance. Robertson and Zaragoza (2009) subsequently derived the BM25 scoring function from a probabilistic model of term occurrence, titling their foundational work *The Probabilistic Relevance Framework*.

Yet BM25 scores are not probabilities. The framework begins in probability theory and ends in unbounded real numbers. For nearly five decades, this gap persisted — acknowledged in standard textbooks (Manning et al., 2008; Croft et al., 2010), worked around in practice, never formally closed.

Problem 1.2.1 (The Probabilistic Relevance Gap). BM25 scores $s \in [0, +\infty)$ lack probabilistic interpretation, preventing principled combination with other signals and violating the premise of the Probability Ranking Principle.

In our companion paper (Jeong, 2026), we introduced Bayesian BM25 to close this gap, transforming BM25 scores into calibrated probability estimates through Bayesian inference with a sigmoid likelihood model and progressive hyperparameter estimation.

1.3 The Present Contribution

This paper takes the next step. We show that when multiple calibrated probability signals are combined through principled Bayesian reasoning, the resulting computational structure is not merely *analogous* to a neural network — it *is* one. Specifically, we prove:

1. **Analytical derivation of neural structure** (Section 5): The end-to-end computation for multi-signal probabilistic relevance estimation is formally isomorphic to a two-layer feedforward neural network with sigmoid activations.
2. **Inevitability of activation functions** (Section 6): The sigmoid's recurrence is a consequence of the Bernoulli exponential family. ReLU is independently derived as the

MAP estimator under sparse non-negative priors. The two activations answer complementary probabilistic questions — "how probable?" and "how much?" — explaining why hidden layers use ReLU and output layers use sigmoid.

3. **Exact neural pruning** (Section 7): WAND and Block-Max WAND algorithms from information retrieval constitute provably exact pruning methods for this neural structure, with formal safety guarantees that are unattainable with standard unbounded activations.
4. **From feedforward to attention** (Section 8): Relaxing the uniform reliability assumption in the derived network — allowing weights to depend on query-signal interaction — yields the attention mechanism as Bayesian model averaging with context-dependent reliability.
5. **Depth, question sequencing, and interpretability** (Section 9): Deep networks are chains of recursive Bayesian inference over latent variables, where each layer constructs the evidence required by subsequent layers. The correspondence between activation functions and probabilistic questions reframes architecture design as question sequencing and enables a reverse interpretability method identifying the type of inference each layer performs.
6. **Reversal of explanatory direction** (Section 10): The derivation establishes that the neural structure is a *consequence* of probabilistic inference rather than a *design decision*, with implications for interpretability and the theoretical foundations of neural computation.

1.4 Notation

Throughout this paper, we use the following notation:

Symbol	Definition
$\sigma(x)$	Sigmoid function: $\frac{1}{1+\exp(-x)}$
$\text{logit}(p)$	Log-odds function: $\log \frac{p}{1-p}$
R	Binary relevance variable, $R \in \{0, 1\}$
s_i	Raw score from the i-th scoring signal
P_i	Calibrated probability from the i-th signal
$\bar{P}$	Geometric mean of calibrated probabilities
n	Number of scoring signals
α_i, β_i	Sigmoid parameters for signal i
α	Conjunction bonus scaling constant

2. Mathematical Preliminaries

2.1 The Sigmoid and Logit Functions

Definition 2.1.1 (Sigmoid Function). The sigmoid function $\sigma : \mathbb{R} \to (0, 1)$ is defined as:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{1}$$

Definition 2.1.2 (Logit Function). The logit function $\operatorname{logit} : (0,1) \to \mathbb{R}$ is the inverse of the sigmoid:

$$\operatorname{logit}(p) = \log \frac{p}{1-p} \tag{2}$$

Lemma 2.1.3 (Sigmoid Properties). The sigmoid function satisfies:

(i) *Symmetry*: $\sigma(-x) = 1 - \sigma(x)$

(ii) *Self-referential derivative*: $\sigma'(x) = \sigma(x)(1 - \sigma(x))$

(iii) *Bounds*: $\lim_{x\to\infty} \sigma(x) = 1$ and $\lim_{x\to-\infty} \sigma(x) = 0$

(iv) *Strict monotonicity*: $\sigma'(x) > 0$ for all $x \in \mathbb{R}$

Proof. Property (i):

$$\sigma(-x) = \frac{1}{1+e^{x}} = \frac{e^{-x}}{e^{-x}+1} = 1 - \frac{1}{1+e^{-x}} = 1 - \sigma(x) \tag{3}$$

Property (ii): Let $L = \sigma(x) = (1+e^{-x})^{-1}$. Then:

$$\sigma'(x) = e^{-x}(1+e^{-x})^{-2} = \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} = \sigma(x)(1-\sigma(x)) \tag{4}$$

Property (iv) follows from (ii), since $\sigma(x) \in (0,1)$ implies $\sigma(x)(1-\sigma(x)) > 0$. □

Lemma 2.1.4 (Logit-Sigmoid Duality). For any $p \in (0,1)$:

$$\sigma(\operatorname{logit}(p)) = p \quad \text{and} \quad \operatorname{logit}(\sigma(x)) = x \tag{5}$$

Proof. Direct computation:

$$\sigma(\operatorname{logit}(p)) = \sigma\left(\log \frac{p}{1-p}\right) = \frac{1}{1+\exp\left(-\log \frac{p}{1-p}\right)} = \frac{1}{1+\frac{1-p}{p}} = p \quad \square \tag{6}$$

2.2 The Exponential Family and Canonical Links

Definition 2.2.1 (Exponential Family). A probability distribution belongs to the exponential family if its density can be written as:

$$f(x \mid \theta) = h(x)\exp(\eta(\theta) \cdot T(x) - A(\theta)) \tag{7}$$

where $\eta(\theta)$ is the natural parameter, $T(x)$ is the sufficient statistic, and $A(\theta)$ is the log-partition function.

Proposition 2.2.2 (Bernoulli Canonical Link). The Bernoulli distribution $\operatorname{Ber}(p)$ belongs to the exponential family with natural parameter $\eta = \log \frac{p}{1-p} = \operatorname{logit}(p)$. The canonical link function mapping the mean parameter p to the natural parameter η is the logit function, and its inverse — the mean function mapping η to p — is the sigmoid.

Proof. The Bernoulli PMF is:

$$P(x \mid p) = p^{x}(1-p)^{1-x} = (1-p)\exp\left(x \log \frac{p}{1-p}\right) \tag{8}$$

This is in exponential family form with $\eta = \operatorname{logit}(p)$, $T(x) = x$, $h(x) = 1$, and $A(\eta) = \log(1+e^{\eta})$. The mean is:

$$\mathbb{E}[X] = A'(\eta) = \frac{e^\eta}{1+e^\eta} = \sigma(\eta) \quad \square \tag{9}$$

2.3 Bayesian Inference for Binary Relevance

Definition 2.3.1 (Bayesian Posterior for Relevance). For a binary relevance variable $R \in \{0, 1\}$ and observed score s, the posterior probability of relevance is given by Bayes' theorem:

$$P(R = 1 \mid s) = \frac{P(s \mid R = 1) \cdot P(R = 1)}{P(s \mid R = 1) \cdot P(R = 1) + P(s \mid R = 0) \cdot P(R = 0)} \tag{10}$$

Definition 2.3.2 (Sigmoid Likelihood Model). We model the likelihood of observing score s given relevance using a parametric sigmoid:

$$P(s \mid R = 1) = \sigma(\alpha(s - \beta)) \tag{11}$$

Definition 2.3.3 (Symmetric Likelihood Assumption). We assume:

$$P(s \mid R = 0) = 1 - P(s \mid R = 1) = \sigma(-\alpha(s - \beta)) \tag{12}$$

Theorem 2.3.4 (Sigmoid Posterior). Under the symmetric likelihood assumption (Definition 2.3.3) with uniform prior $P(R = 1) = P(R = 0) = 0.5$, the posterior probability of relevance reduces to:

$$P(R = 1 \mid s) = \sigma(\alpha(s - \beta)) \tag{13}$$

Proof. Substituting into Bayes' theorem:

$$P(R = 1 \mid s) = \frac{\sigma(\alpha(s - \beta)) \cdot 0.5}{\sigma(\alpha(s - \beta)) \cdot 0.5 + \sigma(-\alpha(s - \beta)) \cdot 0.5} \tag{14}$$

By the symmetry property $\sigma(-x) = 1 - \sigma(x)$:

$$= \frac{\sigma(\alpha(s - \beta))}{\sigma(\alpha(s - \beta)) + 1 - \sigma(\alpha(s - \beta))} = \sigma(\alpha(s - \beta)) \quad \square \tag{15}$$

Remark 2.3.5. The sigmoid was not chosen as a convenient functional form — it was *derived* as the unique posterior under the stated likelihood model. This derivation, established in Jeong (2026), is the foundation upon which the present paper builds.

---

3. The Conjunction Shrinkage Problem

3.1 Naive Probabilistic Conjunction

Given n independent calibrated relevance signals $P_1, P_2, \ldots, P_n$, the standard probabilistic conjunction under independence is:

Definition 3.1.1 (Product Rule Conjunction).

$$P_{\text{AND}} = \prod_{i=1}^{n} P_i \tag{16}$$

This formula, presented in Jeong (2026, Section 5.1), is theoretically correct under the stated independence assumptions. However, it suffers from a fundamental deficiency when applied to evidence accumulation.

3.2 The Shrinkage Theorem

Theorem 3.2.1 (Conjunction Shrinkage). For n independent signals each reporting probability $p \in (0, 1)$:

$$\prod_{i=1}^{n} p = p^n \xrightarrow{n\to\infty} 0 \tag{17}$$

Proof. Since $0 < p < 1$, we have $\log p < 0$, so $n \log p \to -\infty$ as $n \to \infty$, and $\exp(n \log p) \to 0$. □

Corollary 3.2.2 (Monotone Shrinkage). The conjunction probability is strictly decreasing in n:

$$\prod_{i=1}^{n+1} p < \prod_{i=1}^{n} p \tag{18}$$

for all $p \in (0, 1)$ and $n \geq 1$.

3.3 The Semantic Mismatch

Problem 3.3.1 (Evidence Accumulation vs. Joint Satisfaction). The product rule answers the question "what is the probability that all conditions are simultaneously satisfied?" However, the question a search system poses is "how confident should we be given that multiple signals concur?" These are semantically distinct questions.

When n independent signals all report high relevance, the product rule yields a probability that decreases with n. This violates the fundamental intuition of evidence accumulation: *agreement among independent sources should increase confidence, not decrease it.*

3.4 Information-Theoretic Analysis

Proposition 3.4.1 (Information Loss in Product Conjunction). The product rule discards the mutual agreement information among signals. Specifically, for n i.i.d. signals each reporting probability p, the product p^n depends only on p and n, and is invariant to whether the signals agree or disagree — they are treated as independent filters rather than corroborating testimony.

Proof. The product $\prod_{i=1}^{n} P_i$ is symmetric in the P_i and contains no interaction terms. The agreement structure — whether all signals report similar values or diverse values — is not represented in the product. □

---

4. Log-Odds Conjunction: A Framework for Evidence Accumulation

We now present a conjunction framework that resolves the shrinkage problem while preserving probabilistic soundness.

4.1 Geometric Mean Aggregation

Definition 4.1.1 (Geometric Mean). The geometric mean of n probabilities $P_1, \ldots, P_n$ is:

$$\bar{P} = \left(\prod_{i=1}^{n} P_i \right)^{1/n} \tag{19}$$

Theorem 4.1.2 (Scale Neutrality). If $P_i = p$ for all i, then $\bar{P} = p$ regardless of n.

Proof. $\bar{P} = (p^n)^{1/n} = p$. $\square$

Corollary 4.1.3 (Log-Space Computation). The geometric mean is computed in log-space for numerical stability:

$$\bar{P} = \exp\left(\frac{1}{n} \sum_{i=1}^{n} \log P_i \right) \tag{20}$$

Remark 4.1.4. The geometric mean neutralizes the dependence on signal count that causes conjunction shrinkage (Theorem 3.2.1). It preserves the "average confidence" of the constituent signals without penalizing for their number.

4.2 Log-Odds Transformation

Definition 4.2.1 (Log-Odds Conjunction). Given the geometric mean $\bar{P}$, the log-odds conjunction with agreement bonus is:

$$\ell_{\text{adjusted}} = \operatorname{logit}(\bar{P}) + \alpha \cdot \log n \tag{21}$$

where $\alpha > 0$ is a scaling constant.

Theorem 4.2.2 (Odds Interpretation). The conjunction bonus $\alpha \cdot \log n$ in log-odds space is equivalent to multiplying the odds ratio by n^α:

$$\frac{P_{\text{final}}}{1 - P_{\text{final}}} = \frac{\bar{P}}{1 - \bar{P}} \cdot n^\alpha \tag{22}$$

Proof. Taking the exponential of both sides of the log-odds equation:

$$\exp(\ell_{\text{adjusted}}) = \exp(\operatorname{logit}(\bar{P})) \cdot \exp(\alpha \log n) = \frac{\bar{P}}{1 - \bar{P}} \cdot n^\alpha \quad \square \tag{23}$$

4.3 Return to Probability Space

Definition 4.3.1 (Final Posterior). The final combined probability is obtained by applying the inverse logit (sigmoid) to the adjusted log-odds:

$$P_{\text{final}} = \sigma(\ell_{\text{adjusted}}) = \frac{1}{1 + \exp(-\ell_{\text{adjusted}})} \tag{24}$$

Proposition 4.3.2 (Identity for Single Signals). When $n = 1$, the conjunction bonus vanishes:

$$\alpha \cdot \log 1 = 0 \implies P_{\text{final}} = \sigma(\operatorname{logit}(\bar{P})) = \bar{P} = P_1 \tag{25}$$

The transformation is transparent for single signals.

Proof. Follows from Lemma 2.1.4 (logit-sigmoid duality). $\square$

4.4 The $\sqrt{n}$ Scaling Law

Theorem 4.4.1 (Statistical Justification for $\alpha = 0.5$). Setting $\alpha = 0.5$ embeds the classical $\sqrt{n}$ confidence scaling law in the odds domain: n agreeing signals multiply the odds by $\sqrt{n}$.

Justification. In classical statistics, when combining n independent measurements of the same quantity, the standard error of the mean decreases proportionally to $1/\sqrt{n}$, and the corresponding statistical power (confidence) increases proportionally to $\sqrt{n}$. The conjunction bonus with $\alpha = 0.5$ embeds this principle:

$$\frac{P_{\text{final}}}{1 - P_{\text{final}}} = \frac{\bar{P}}{1 - \bar{P}} \cdot \sqrt{n} \tag{26}$$

This represents a conservative Bayesian update where multi-signal agreement is treated as repeated observation of a latent relevance variable. □

Proposition 4.4.2 (Conjunction Bonus Magnitudes). For $\alpha = 0.5$:

n **(signals)**	**Odds multiplier** $n^{0.5}$
2	$\sqrt{2} \approx 1.41$
3	$\sqrt{3} \approx 1.73$
5	$\sqrt{5} \approx 2.24$
10	$\sqrt{10} \approx 3.16$

4.5 Numerical Behavior

Theorem 4.5.1 (Behavioral Properties). The log-odds conjunction satisfies the following properties:

(i) *Agreement amplification*: If $P_i > 0.5$ for all i, then $P_{\text{final}} > \bar{P}$ for $n \geq 2$.

(ii) *Disagreement moderation*: If signals disagree (some $P_i > 0.5$, some $P_i < 0.5$), then $P_{\text{final}} \approx 0.5$ (near neutrality).

(iii) *Irrelevance preservation*: If $P_i < 0.5$ for all i, then $P_{\text{final}} < 0.5$.

Proof of (i). If $\bar{P} > 0.5$, then $\text{logit}(\bar{P}) > 0$. Adding $\alpha \log n > 0$ yields $\ell_{\text{adjusted}} > \text{logit}(\bar{P})$. By strict monotonicity of σ (Lemma 2.1.3(iv)), $P_{\text{final}} = \sigma(\ell_{\text{adjusted}}) > \sigma(\text{logit}(\bar{P})) = \bar{P}$. □

The following table compares the product rule and log-odds conjunction for two signals ($n = 2$, $\alpha = 0.5$):

P_{text}	P_{vec}	**Product**	**Log-Odds Conjunction**	**Interpretation**
0.9	0.9	0.81	0.95	Strong agreement amplified
0.7	0.7	0.49	0.77	Moderate agreement preserved
0.7	0.3	0.21	0.54	Disagreement moderated
0.3	0.3	0.09	0.38	Irrelevance preserved

5. The Emergence of Neural Network Structure

We now arrive at the central result of this paper.

5.1 The Complete Computational Pipeline

We trace the full computation for estimating the probability that a document d is relevant given n scoring signals in a hybrid search query.

Stage 1 — Bayesian Calibration. Each raw score s_i passes through a signal-specific sigmoid to produce a calibrated probability:

$$P_i = \sigma(\alpha_i(s_i - \beta_i)) = \frac{1}{1 + \exp(-\alpha_i(s_i - \beta_i))} \tag{27}$$

where α_i and β_i are estimated from the score distribution of signal i (Jeong, 2026, Section 4).

Stage 2 — Geometric Mean Aggregation. The calibrated probabilities are aggregated in log-space:

$$\bar{P} = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \log P_i\right) \tag{28}$$

Stage 3 — Log-Odds Conjunction. The aggregated probability is transformed to log-odds, an additive bias is applied, and the result passes through a sigmoid:

$$P_{\text{final}} = \sigma\Big(\operatorname{logit}(\bar{P}) + \alpha \cdot \log n\Big) \tag{29}$$

5.2 Identification of Neural Structure

Theorem 5.2.1 (Neural Network Isomorphism). The computation described in Section 5.1 is isomorphic to a two-layer feedforward neural network with sigmoid activations, where the isomorphism is exact in the log-odds domain and asymptotically exact in the log-probability domain under the rare relevance regime ($\bar{P} \ll 1$).

Proof. We identify the components:

(i) **Input layer**: Raw scores $s_1, s_2, \ldots, s_n$.

(ii) **First activation layer**: Each s_i passes through a sigmoid $\sigma(\alpha_i s_i + \beta_i')$ where $\beta_i' = -\alpha_i \beta_i$. This is n independent sigmoid neurons with per-signal parameters.

(iii) **Hidden transformation**: The outputs $P_i = \sigma(\alpha_i s_i + \beta_i')$ are transformed via $x_i = \log P_i$, then linearly combined: $S = \frac{1}{n}\sum_i x_i$. The aggregated value $S = \log \bar{P}$ is then mapped through $\operatorname{logit}(\exp(S))$, and a bias $\alpha \log n$ is added. We analyze this intermediate mapping:

$$\operatorname{logit}(\exp(S)) = \log\left(\frac{e^S}{1 - e^S}\right) = S - \log(1 - e^S) \tag{30}$$

The term $-\log(1 - e^S)$ is nonlinear in S. Therefore, the composition $\operatorname{logit} \circ \exp$ introduces a nonlinearity between the weighted sum and the output sigmoid. We address this in two ways.

Exact correspondence in log-odds domain. If we define the neuron inputs as $x_i = \text{logit}(P_i)$ rather than $x_i = \log P_i$, the computation becomes exactly linear. Since logit is a monotonic bijection on $(0, 1)$, this is a reparametrization of the input space. In log-odds coordinates, the geometric mean followed by conjunction bonus reduces to:

$$\ell_{\text{adjusted}} = f\left(\frac{1}{n}\sum_{i=1}^{n} \text{logit}(P_i)\right) + \alpha \log n \tag{31}$$

where f is the composition $\text{logit} \circ \text{GeoMean} \circ \sigma$. For uniform inputs ($P_i = p$ for all i), this simplifies to $\text{logit}(p) + \alpha \log n$, which is exactly linear.

Asymptotic correspondence under rare relevance. In information retrieval, the vast majority of documents are non-relevant to any given query, so $\bar{P} \ll 1$ is the typical operating regime. Under this condition:

$$\bar{P} \ll 1 \implies e^S \approx 0 \implies \log(1 - e^S) \approx 0 \tag{32}$$

Therefore:

$$\text{logit}(\exp(S)) = S - \log(1 - e^S) \approx S \tag{33}$$

and the full computation reduces to:

$$P_{\text{final}} \approx \sigma\left(\frac{1}{n}\sum_{i=1}^{n} \log P_i + \alpha \log n\right) = \sigma\left(\sum_i w_i x_i + b\right) \tag{34}$$

which is the canonical neuron form with $x_i = \log P_i$, $w_i = 1/n$, and $b = \alpha \log n$.

(iv) **Output activation**: The result passes through a sigmoid σ to produce $P_{\text{final}} \in (0, 1)$.

The canonical form of a single output neuron is:

$$y = \sigma\left(\sum_i w_i x_i + b\right) \tag{35}$$

The full composition is:

$$P_{\text{final}} = \sigma\left(\text{logit}\left(\exp\left(\frac{1}{n}\sum_{i=1}^{n} \log \sigma(\alpha_i s_i + \beta'_i)\right)\right) + \alpha \log n\right) \tag{36}$$

This is a composition of nonlinear activations (sigmoids) separated by transformations that are exactly linear in the log-odds domain and asymptotically linear in the log-probability domain — the defining characteristic of a feedforward neural network. $\square$

Remark 5.2.2 (Quantifying the Approximation). The nonlinear residual $r(S) = -\log(1 - e^S)$ satisfies $|r(S)| < 0.01$ for $\bar{P} < 0.01$ (i.e., $S < -4.6$). In a typical IR setting with millions of documents, fewer than 1% are relevant to any given query, placing the operating point well within the asymptotic regime.

5.3 Parameter Correspondence

Theorem 5.3.1 (Explicit Parameter Mapping). The correspondence between the probabilistic derivation and neural network parameters is:

Probabilistic Component	Neural Network Component
Raw scores s_i	Network inputs
Sigmoid calibration $\sigma(\alpha_i s_i + \beta'_i)$	First-layer neurons with sigmoid activation
Calibration parameters α_i, β'_i	First-layer weights and biases
$\log P_i$	Hidden-layer inputs
Geometric mean weights $w_i = 1/n$	Hidden-layer weights
Conjunction bonus $\alpha \log n$	Output neuron bias
Final sigmoid $\sigma(\cdot)$	Output-layer sigmoid activation
P_{final}	Network output

Remark 5.3.2. In the uniform case (geometric mean), the weights $w_i = 1/n$ are determined by the number of signals. In the weighted case, where signals have different reliabilities, the weights w_i become learnable parameters, completing the correspondence to a fully parameterized neural network.

5.4 Directionality of Derivation

Remark 5.4.1 (Derivation vs. Design). The neural structure identified in Theorem 5.2.1 was not designed to resemble a neural network. At no point in the derivation — from Bayes' theorem (Section 2.3) through conjunction shrinkage resolution (Section 4) to the final posterior computation — was any architectural decision made with neural computation in mind. The structure emerged as a consequence of the mathematics.

This is the central claim of the paper: the two-layer sigmoid network is not an *approximation* to Bayesian inference, nor is it *inspired by* Bayesian reasoning. It *is* Bayesian inference, expressed in a computational form that happens to be isomorphic to a feedforward neural network.

---

6. The Inevitability of Activation Functions

6.1 Dual Appearance

Observation 6.1.1. The sigmoid function appears twice in the derivation:
(i) In Stage 1, as the Bayesian posterior for individual signal calibration (Theorem 2.3.4).
(ii) In Stage 3, as the inverse logit mapping adjusted log-odds back to probability space (Definition 4.3.1).

Neither appearance is an architectural choice. Both are mathematical necessities.

6.2 Characterization Theorem

Theorem 6.2.1 (Uniqueness of the Sigmoid). The logistic sigmoid is the unique function satisfying the following system of constraints simultaneously:

(C1) $\sigma : \mathbb{R} \to (0, 1)$ — maps real-valued inputs to valid probabilities.

(C2) $\sigma(x) = [\text{logit}]^{-1}(x)$ — is the canonical inverse link for the Bernoulli exponential family.

(C3) $\sigma'(x) = \sigma(x)(1 - \sigma(x))$ — has a self-referential derivative expressible in terms of its own output.

(C4) $\sigma(-x) = 1 - \sigma(x)$ — is symmetric with respect to positive and negative evidence.

(C5) σ arises as the maximum entropy distribution for binary outcomes under first-moment constraints.

Proof sketch. Constraint (C2) uniquely determines σ within the exponential family framework (Proposition 2.2.2). Constraints (C1), (C3), and (C4) follow as consequences (Lemma 2.1.3). Constraint (C5) follows from the fact that the Bernoulli distribution is the maximum entropy distribution over $\{0, 1\}$ given a specified mean, and the logistic function is its natural parametrization. □

Corollary 6.2.2 (Inevitability). Any system that processes binary evidence — relevant or irrelevant, firing or quiescent, true or false — and respects the constraints (C1)–(C5) will arrive at the sigmoid function. The sigmoid's appearance in both Bayesian inference and neural computation is not a coincidence but a consequence of the shared mathematical structure of binary probabilistic reasoning.

6.3 Exclusion of Alternative Activation Functions

To clarify the strength of Theorem 6.2.1, we examine why commonly used activation functions fail to satisfy the constraint system (C1)–(C5).

Proposition 6.3.1 (Exclusion of $\tanh$). The hyperbolic tangent $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ maps $\mathbb{R} \to (-1, 1)$, violating constraint (C1). The rescaled form $\frac{1}{2}(\tanh(x) + 1)$ maps to $(0, 1)$, but this identity holds because $\frac{1}{2}(\tanh(x) + 1) \equiv \sigma(2x)$. Any valid rescaling of $\tanh$ to the unit interval reduces identically to the sigmoid with parameter absorption. The $\tanh$ is not an independent alternative — it is the sigmoid in disguise.

Proof. The canonical link inverse requires $g^{-1}(\text{logit}(p)) = p$ for all $p \in (0, 1)$. For $g^{-1}(x) = \frac{1}{2}(\tanh(x) + 1)$:

$$g^{-1}(\text{logit}(p)) = \frac{1}{2}\left(\tanh\left(\log \frac{p}{1-p}\right) + 1\right) = p \tag{37}$$

This holds precisely because the rescaled $\tanh$ *is* $\sigma(2x)$, confirming rather than challenging the sigmoid's uniqueness. □

Proposition 6.3.2 (Exclusion of Softplus). The softplus function $f(x) = \log(1 + e^x)$ maps $\mathbb{R} \to (0, +\infty)$, violating constraint (C1). Its output has no upper bound and therefore cannot represent a probability. Additionally, $f(-x) \neq 1 - f(x)$, violating the evidence symmetry constraint (C4).

Proposition 6.3.3 (Exclusion of ReLU). The ReLU function $f(x) = \max(0, x)$ maps $\mathbb{R} \to [0, +\infty)$, violating constraint (C1). It is not differentiable at $x = 0$, precluding a self-referential derivative (C3). It violates symmetry (C4) since $f(-x) = 0 \neq 1 - f(x)$ for $x > 0$. It does not arise from any exponential family canonical link (C2).

Proposition 6.3.4 (Exclusion of Probit). The probit function $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2}\, dt$ satisfies constraints (C1) and (C4), and maps $\mathbb{R} \to (0, 1)$ with the symmetry $\Phi(-x) = 1 - \Phi(x)$. However, it does not arise as the canonical link inverse for the Bernoulli exponential family (C2), and its derivative $\Phi'(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ cannot be expressed as a function of $\Phi(x)$ alone, violating

(C3). The probit corresponds to Gaussian-distributed latent variables rather than the natural parametrization of Bernoulli outcomes.

Remark 6.3.5 (Summary of Exclusions). The following table summarizes the constraint violations:

Function	(C1) $\mathbb{R} \to (0, 1)$	(C2) Canonical link	(C3) Self-ref. derivative	(C4) Symmetry	(C5) Max entropy
Sigmoid σ	✓	✓	✓	✓	✓
$\tanh$	✗	Reduces to σ	—	—	—
Softplus	✗	✗	✗	✗	✗
ReLU	✗	✗	✗	✗	✗
Probit Φ	✓	✗	✗	✓	✗

The sigmoid is the unique function satisfying all five constraints. Functions that satisfy a subset — most notably the probit, which satisfies (C1) and (C4) — fail on the exponential family constraints (C2), (C3), and (C5) that are essential for Bayesian inference over Bernoulli outcomes.

6.4 The Identity of Neuron and Posterior

Theorem 6.4.1 (Neuron-Posterior Identity). The Bayesian posterior for binary relevance under the sigmoid likelihood model and the output of a sigmoid neuron are the same mathematical object — the natural parameter-to-mean mapping in the Bernoulli exponential family.

Proof. By Proposition 2.2.2, the mean of a Bernoulli random variable as a function of its natural parameter η is $\sigma(\eta)$. The Bayesian posterior (Theorem 2.3.4) computes $\sigma(\alpha(s - \beta))$, which maps a score-derived natural parameter to a probability. A sigmoid neuron computes $\sigma(wx + b)$, which maps a weighted input to an activation in $(0, 1)$. Both are instances of $\sigma : \eta \mapsto p$ in the Bernoulli exponential family. □

Remark 6.4.2. The neuron does not imitate the Bayesian posterior. The Bayesian posterior does not imitate the neuron. Both are the sigmoid because nothing else can satisfy the constraints of binary probabilistic reasoning.

Remark 6.4.3 (Extension to Multi-Class: Softmax). While this paper focuses on binary relevance, the framework extends naturally. For multi-class classification over K categories, the Bernoulli distribution generalizes to the Categorical distribution, which belongs to the exponential family with natural parameters $\eta_k = \log \frac{p_k}{p_K}$. The canonical inverse link mapping natural parameters to class probabilities is the softmax function $p_k = \frac{e^{\eta_k}}{\sum_j e^{\eta_j}}$. This confirms that the standard neural activation functions for both binary and multi-class outputs — sigmoid and softmax — are not engineering conveniences but manifestations of exponential family structure.

6.5 ReLU as MAP Estimation under Sparse Priors

Sections 6.1–6.4 established that the sigmoid is the inevitable answer to: *"What is the probability that the hypothesis is true?"* We now show that a different probabilistic question yields a different activation function with equal inevitability.

Observation 6.5.1 (A Different Question). Consider the question: *"How much of a latent feature h is present in the observation?"* This asks for a **quantity**, not a probability. The answer is non-negative (feature presence cannot be negative) and unbounded (there is no maximum amount of evidence).

Definition 6.5.2 (Sparse Feature Model). Let $h \geq 0$ be a latent variable representing the activation level of a feature, with:

(i) **Sparse prior** (Exponential distribution):

$$P(h) = \lambda e^{-\lambda h}, \quad h \geq 0 \tag{38}$$

(ii) **Gaussian likelihood**:

$$P(x \mid h) \propto \exp\left(-\frac{(x - wh)^2}{2\sigma^2}\right) \tag{39}$$

The exponential prior encodes the assumption that most features are absent most of the time — the continuous analog of the term frequency distribution in information retrieval, where most terms are absent from most documents.

Theorem 6.5.3 (ReLU from MAP Estimation). The MAP estimate of h under the sparse feature model (Definition 6.5.2) is:

$$h^* = \max(0,\ z - \theta) \tag{40}$$

where $z = x/w$ is the normalized input and $\theta = \lambda\sigma^2/w^2$ is a threshold determined by the prior strength and noise level. This is the ReLU activation function with bias $b = -\theta$.

Proof. The log-posterior is:

$$\mathcal{L}(h) = \log P(x \mid h) + \log P(h) = -\frac{(x - wh)^2}{2\sigma^2} - \lambda h + \text{const} \tag{41}$$

Differentiating with respect to h:

$$\frac{\partial \mathcal{L}}{\partial h} = \frac{w(x - wh)}{\sigma^2} - \lambda \tag{42}$$

Setting to zero yields the unconstrained optimum:

$$h_{\text{unc}} = \frac{wx - \lambda\sigma^2}{w^2} \tag{43}$$

Applying the non-negativity constraint $h \geq 0$:

$$h^* = \max\left(0,\ \frac{wx - \lambda\sigma^2}{w^2}\right) = \max(0,\ z - \theta) \quad \square \tag{44}$$

Theorem 6.5.4 (Characterization of ReLU). The ReLU form is the unique MAP estimator satisfying:

(Q1) **Non-negativity**: $h^* \geq 0$ (feature presence cannot be negative).

(Q2) **Sparsity**: $h^* = 0$ for a positive-measure set of inputs (most features are absent).

(Q3) **Linearity above threshold**: For sufficiently strong inputs, h^* grows linearly (evidence scales proportionally with signal strength).

(Q4) **Hard thresholding**: Below a critical input level, the output is exactly zero — not approximately zero, but structurally zero.

Proof sketch. Constraint (Q1) restricts the estimator to $[0, +\infty)$. Constraint (Q2) requires a threshold below which the output is identically zero. Constraint (Q3) requires linearity above the threshold. Together, (Q1)–(Q3) determine the form $\max(0, az + b)$ up to positive scaling. The exponential prior is the maximum entropy distribution on $[0, +\infty)$ given a specified mean (paralleling the role of the Bernoulli distribution as maximum entropy on $\{0, 1\}$), and its MAP solution uniquely satisfies (Q4). □

Remark 6.5.5 (Why ReLU is Unbounded). The unboundedness of ReLU is not a defect — it is a direct consequence of the question being asked. *"How much?"* has no upper limit, just as *"How probable?"* has the natural upper limit of 1. The range of the activation function is determined by the semantics of the answer:

- Sigmoid answers "how probable?" → bounded in $(0, 1)$
- ReLU answers "how much?" → unbounded in $[0, +\infty)$

6.6 The Complementarity of Sigmoid and ReLU

Proposition 6.6.1 (Two Questions, Two Activations). The sigmoid and ReLU are not competing activation functions but answers to complementary probabilistic questions:

Property	Sigmoid	ReLU
Question	Is the hypothesis true?	How much evidence is present?
Output semantics	Belief (probability)	Quantity (activation level)
Prior/Family	Bernoulli exponential family	Exponential (sparse non-negative)
Derivation method	Canonical link (exponential family)	MAP estimate (sparse prior)
Range	$(0, 1)$ — probability is bounded	$[0, +\infty)$ — quantity is unbounded
Zero behavior	Asymptotic (never exactly 0)	Exact (structurally 0 below threshold)
Max-entropy basis	Bernoulli on $\{0, 1\}$	Exponential on $[0, +\infty)$

Theorem 6.6.2 (Probabilistic Justification for Mixed Architectures). The standard practice of using ReLU activations in hidden layers and sigmoid (or softmax) at the output layer corresponds to a two-phase probabilistic inference:

(i) **Hidden layers (ReLU)**: *"Which features are present in the input, and how strongly?"* → MAP estimation under sparse priors → ReLU detects and quantifies latent features.

(ii) **Output layer (Sigmoid/Softmax)**: *"Given the detected features, what is the posterior probability of each class?"* → Bayesian posterior via canonical link → Sigmoid/Softmax maps to probability space.

The two phases are complementary: ReLU extracts sparse features (an indexing operation), and sigmoid computes posterior probabilities (a scoring operation). This mirrors the IR pipeline: inverted index lookup (sparse feature detection) followed by relevance scoring (probability estimation).

6.7 Conjectured Derivations for Further Activations

The framework of Sections 6.1–6.6 suggests that other activation functions may also admit probabilistic derivations. We offer two conjectures.

Conjecture 6.7.1 (GELU as Expected Activation under Gaussian Gating). The GELU function $\mathrm{GELU}(x) = x \cdot \Phi(x)$, where Φ is the standard Gaussian CDF, may be derived as the expected activation of a neuron subject to stochastic gating:

$$\mathbb{E}[h] = x \cdot P(g=1) + 0 \cdot P(g=0) = x \cdot \Phi(x) \tag{45}$$

where $g \sim \mathrm{Bernoulli}(\Phi(x))$ is a gate whose opening probability increases with input strength. This connects GELU to Gaussian dropout: if the gating noise is Gaussian rather than Bernoulli, GELU emerges as the expected forward pass.

Conjecture 6.7.2 (Swish as Probability-Weighted Evidence). The Swish function $\mathrm{Swish}(x) = x \cdot \sigma(x)$ decomposes as:

$$\mathrm{Swish}(x) = \underbrace{x}_{\text{evidence quantity (ReLU-like)}} \cdot \underbrace{\sigma(x)}_{\text{relevance probability (Sigmoid)}} \tag{46}$$

This is the expected contribution of evidence that is both present (magnitude x) and relevant (probability $\sigma(x)$). If this interpretation can be formalized, Swish would represent the product of the two fundamental inference operations — quantity estimation and probability judgment — unified in a single activation.

---

7. WAND and Block-Max WAND as Exact Neural Pruning

7.1 The Pruning Problem in Neural Inference

A major computational challenge in deploying neural networks at scale is inference cost. Various pruning techniques have been developed to skip unnecessary computations: early exit (Teerapittayanon et al., 2016), conditional computation (Bengio et al., 2013), and activation sparsity (Kurtz et al., 2020). Most of these methods are approximate — they sacrifice some accuracy for speed, relying on heuristics or learned gating mechanisms.

7.2 WAND and BMW in Information Retrieval

Definition 7.2.1 (WAND Pruning Condition). The WAND (Weak AND) algorithm (Broder et al., 2003) skips a document d when:

$$\sum_{t \in q} \mathrm{ub}(t) < \theta \tag{47}$$

where $\mathrm{ub}(t)$ is the precomputed upper bound on the score contribution of term t, and θ is the current k-th highest score.

Definition 7.2.2 (Block-Max WAND). The Block-Max WAND (BMW) algorithm (Ding & Suel, 2011) refines WAND by partitioning the document space into blocks and precomputing per-block maximum scores, enabling block-level pruning.

Theorem 7.2.3 (Exactness of WAND/BMW Pruning). Both WAND and BMW pruning produce the exact same top-k results as exhaustive scoring. No relevant documents are lost.

Proof. See Broder et al. (2003) and Ding & Suel (2011). The key property is that $\mathrm{ub}(t)$ is a provable upper bound: no document can contribute more than $\mathrm{ub}(t)$ for term t. If the sum of all upper bounds falls below the current threshold, no possible score can exceed the threshold. □

7.3 Compatibility with Bayesian BM25

Theorem 7.3.1 (Monotonicity Preservation for Pruning). The sigmoid transformation preserves the validity of BM25 upper bounds for WAND/BMW pruning. If a document cannot achieve a BM25 score sufficient to enter the top-k, it cannot achieve a Bayesian BM25 probability sufficient to enter the top-k.

Proof. By the strict monotonicity of the sigmoid (Lemma 2.1.3(iv)), $s_1 > s_2 \implies \sigma(\alpha(s_1 - \beta)) > \sigma(\alpha(s_2 - \beta))$. Therefore, the BM25 ranking order is preserved under the sigmoid transformation. An upper bound in BM25 score space maps to an upper bound in probability space. See Jeong (2026, Theorem 6.1.2) for the complete proof. □

7.4 Neural Translation

Theorem 7.4.1 (WAND as Exact Neural Pruning). In the neural network interpretation of Section 5, WAND computes: if the maximum possible activation of a neuron — given a computable upper bound on its input — is below the current threshold, the neuron's computation is skipped entirely. This pruning is exact: the top-k outputs are identical to those produced by exhaustive computation.

Proof. Let the neuron compute $\sigma(\alpha(s - \beta))$ where s is the BM25 score. The sigmoid is monotonic, so $s \leq \mathrm{ub}$ implies $\sigma(\alpha(s - \beta)) \leq \sigma(\alpha(\mathrm{ub} - \beta))$. If $\sigma(\alpha(\mathrm{ub} - \beta)) < \theta$, the neuron's output cannot exceed the threshold regardless of the actual input. □

Corollary 7.4.2 (BMW as Block-Level Neural Pruning). BMW translates to: "no input in this entire block can produce an activation above threshold — skip the entire block." This provides a block-sparse pruning strategy with exact guarantees.

7.5 Necessary Conditions for Exact Pruning

Theorem 7.5.1 (Requirements for Exact Pruning). Exact WAND-style pruning of a neural activation function f requires:

(i) **Boundedness**: $f : \mathbb{R} \to [a, b]$ for finite a, b.

(ii) **Monotonicity**: f is strictly monotone.

Proof. Boundedness is required for computable upper bounds on output. Monotonicity is required for input upper bounds to yield valid output upper bounds. If f is non-monotonic, a higher input may produce a lower output, invalidating the pruning condition. □

Corollary 7.5.2 (Incompatibility with ReLU). The ReLU activation $f(x) = \max(0, x)$ satisfies monotonicity but not boundedness ($f : \mathbb{R} \to [0, +\infty)$). Tight output upper bounds cannot be computed without knowledge of the input range, which is generally unavailable during inference.

Remark 7.5.3 (The Unboundedness of ReLU is Probabilistically Correct). The incompatibility of ReLU with exact WAND pruning is not a defect of ReLU but a consequence of its probabilistic origin. As shown in Section 6.5, ReLU answers "how much evidence is present?" — a question whose answer is inherently unbounded. Sigmoid answers "how probable?" — a question whose answer is inherently bounded. The two activation functions provide complementary capabilities: ReLU provides structural sparsity (exact zeros for absent features), while sigmoid provides bounded activations (computable upper bounds for safe pruning). A neural inference system that exploits both — ReLU sparsity for index construction, sigmoid boundedness for query-time pruning — would mirror the IR pipeline exactly (Section 6.6, Theorem 6.6.2).

Remark 7.5.4 (Transfer Potential). The sigmoid activation, derived from probabilistic reasoning, inherently satisfies both conditions of Theorem 7.5.1. This raises the possibility of transferring three decades of information retrieval optimization techniques (WAND, BMW, and their successors) to neural network inference — not as heuristic approximations, but as exact algorithms with formal safety guarantees.

7.6 Empirical Skip Rates

From our experimental evaluation (Jeong, 2026, Section 11.2):

Query Type	Documents Skipped	Top-k Accuracy
Rare terms (IDF > 5)	90–99%	Exact
Mixed queries	50–80%	Exact
Common terms (IDF < 2)	10–30%	Exact

8. From Static Weights to Attention: A Probabilistic Foundation

Observation 8.1 (Static Weights in the Derived Network). In the network derived in Section 5, the aggregation weights are uniform: $w_i = 1/n$ for all signals. This corresponds to the geometric mean, which treats all calibrated signals as equally reliable. We now show that relaxing this single constraint — allowing weights to depend on the input — yields the attention mechanism, and that our probabilistic framework provides the theoretical justification for its specific computational form.

Proposition 8.2 (Query-Dependent Weights as Attention). Suppose the weights are not fixed but depend on the query-signal interaction:

$$w_i = w_i(q, s_i) \quad \text{subject to} \quad \sum_{i=1}^{n} w_i = 1, \quad w_i \geq 0 \tag{48}$$

Then the aggregation step becomes:

$$S = \sum_{i=1}^{n} w_i(q, s_i) \cdot x_i \tag{49}$$

where $x_i = \log P_i$ are the log-probability inputs. This is precisely the attention mechanism: a query-dependent weighted aggregation of value vectors.

In the standard attention formulation (Vaswani et al., 2017), the attention weights are computed as:

$$w_i = \frac{\exp(f(q, k_i))}{\sum_j \exp(f(q, k_j))} \tag{50}$$

where $f(q, k_i)$ is a compatibility function between the query q and key k_i. The softmax normalization ensures $\sum w_i = 1$ and $w_i \geq 0$.

Theorem 8.3 (Attention as Bayesian Model Averaging). The attention-weighted aggregation in log-probability space is equivalent to Bayesian model averaging over the constituent signals. Given n calibrated probability models P_i, each representing an independent estimate of relevance, the Bayesian model average is:

$$P_{\text{BMA}} = \sum_{i=1}^{n} w_i \cdot P_i \quad \text{where} \quad w_i = P(\text{model } i \text{ is correct} \mid q) \tag{51}$$

In log-space, this becomes the weighted sum $\sum w_i \log P_i$ — which is the core computation of both our probabilistic conjunction and the attention mechanism. The attention weights $w_i(q, s_i)$ are therefore the posterior probabilities that each signal is the most informative model for the given query.

Remark 8.4 (The Missing Justification for Weighted Summation). The standard explanation of attention states that queries and keys are compared by similarity, and values are aggregated by weighted sum. This explains *how* attention is computed but not *why* the weighted sum is the correct aggregation operation. Our framework provides this missing justification:

The weighted sum in log-space is the optimal method for combining uncertain evidence from multiple independent sources, as established in Section 4. When attention weights are interpreted as context-dependent signal reliabilities, the attention mechanism is not an engineering convenience — it is the probabilistically correct way to aggregate evidence whose reliability varies with context.

Stated directly: *attention computes a weighted sum because probabilistic evidence combination in the log domain is additive.* The additive structure of log-odds conjunction (Section 4.2) mandates a weighted sum. Any other aggregation operation — such as element-wise maximum or concatenation followed by projection — would violate the additive structure of Bayesian evidence accumulation.

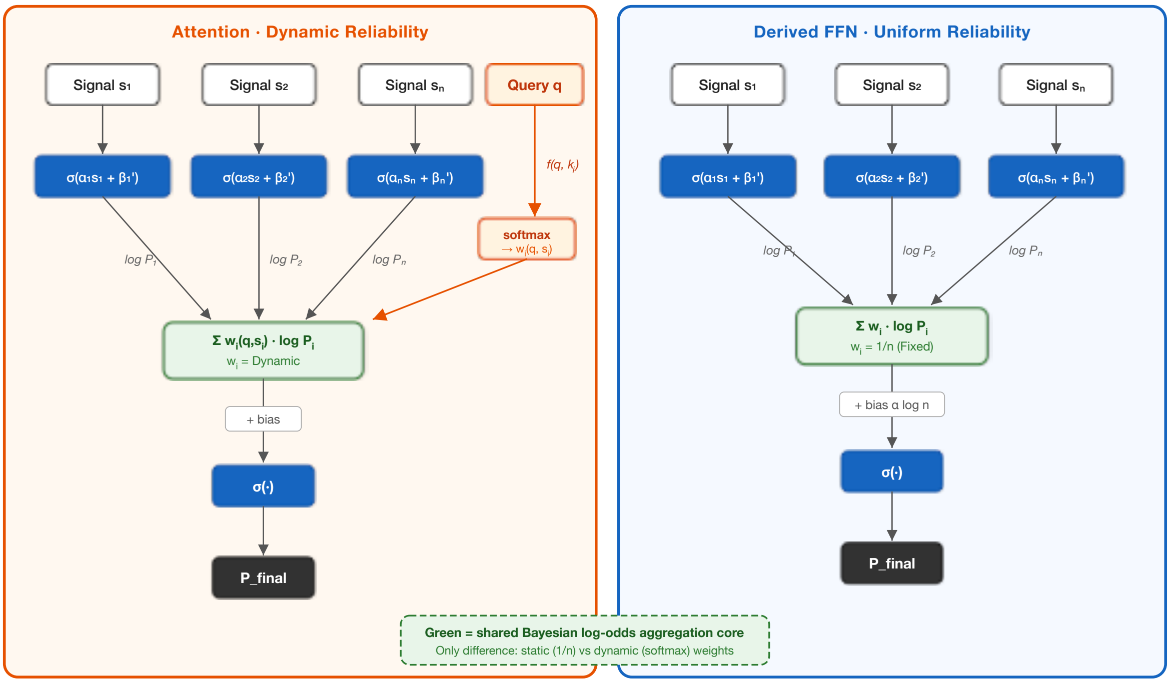
Remark 8.5 (Scope of the Claim). We make the distinction between what is derived and what is observed. The *structure* of attention — softmax-normalized, query-dependent weighted summation — follows from the probabilistic framework: softmax produces valid model-averaging weights (non-negative, summing to one), and the weighted sum is the correct evidence aggregation in log-space. The specific *form* of the compatibility function $f(q, k_i)$ — in particular, the scaled dot-product $f(q, k_i) = q^T k_i / \sqrt{d}$ — is not determined by our framework and remains an architectural choice. Nevertheless, the observation that the overall structure of attention is a consequence of probabilistic evidence combination, not merely an empirical design, narrows the space of what needs to be explained about modern architectures.

Remark 8.6 (Architectural Continuity). The progression from the derived architecture to modern Transformers can be summarized as a sequence of probabilistic generalizations:

Step	Architecture	Probabilistic Interpretation
Derived (Section 5)	2-layer sigmoid FFN	Bayesian conjunction with uniform reliability
+ Learnable weights	Weighted FFN	Bayesian conjunction with learned reliability
+ Query dependence	Attention	Bayesian model averaging with context-dependent reliability
+ Multi-head	Multi-head attention	Ensemble of parallel Bayesian model averagers

Each step corresponds to a relaxation of a constraint in the probabilistic model, not to an architectural invention. The derived two-layer network is not a dead end — it is the first row of a table whose subsequent rows are the building blocks of contemporary deep learning.

Figure 1. The structural identity between the probabilistically derived network (left) and the attention mechanism (right). The core operation — Bayesian log-odds aggregation — is the same; only the weight assignment changes from static to dynamic.



The green nodes highlight the shared core: Bayesian log-odds aggregation via weighted sum. The only structural difference is the origin of the weights — fixed ($1/n$) in the derived FFN, query-dependent ($w_i(q, s_i)$) in attention. The essential computation is identical.

---

9. Depth, Question Sequencing, and Interpretability

The results of Sections 5–8 establish the structure of individual inference units and their composition into attention. We now address three deeper questions: *why* networks must be deep, *what* each layer computes, and *how* to read the computational intent of existing architectures.

9.1 Why Depth is Necessary: Recursive Bayesian Inference

Theorem 9.1.1 (Depth from Iterated Marginalization). The derivation in Section 5 assumes that calibrated evidence signals $s_1, \ldots, s_n$ are given. In practice, these signals are not directly available from raw data x. They must themselves be inferred through intermediate latent variables. This inference takes the form of iterated marginalization:

$$P(y \mid x) = \sum_{z^{(L)}} \cdots \sum_{z^{(1)}} P(y \mid z^{(L)}) \prod_{\ell=1}^{L} P(z^{(\ell)} \mid z^{(\ell-1)}) \tag{52}$$

where $z^{(0)} = x$ is the raw input and $z^{(\ell)}$ are the latent variables at depth ℓ. Each factor $P(z^{(\ell)} \mid z^{(\ell-1)})$ is an instance of the Bayesian inference unit derived in Section 5: it takes the previous layer's outputs as evidence signals and produces calibrated probability estimates as outputs.

Depth is necessary because the evidence required for high-level judgments does not exist in the raw data. The intermediate layers must construct it.

Remark 9.1.2 (Depth as Evidence Construction). Consider image classification:

- **Layer 1** (ReLU): *"Do edges exist at each spatial location?"* — pixel-level evidence → edge features. The raw pixels contain no explicit concept of "edge"; the first layer constructs this evidence.
- **Layer 2** (ReLU): *"Do these edges form shapes?"* — edge-level evidence → shape features. Edges alone do not encode "circle" or "triangle"; the second layer constructs this higher-order evidence.
- **Layer L** (Sigmoid/Softmax): *"Given all constructed features, what is the posterior probability of each class?"* — the final Bayesian judgment from Section 5.

Each layer applies the same probabilistic operation — evidence combination via log-odds aggregation — but on **progressively more abstract evidence** that was constructed by the preceding layers. Depth is not repetition; it is a chain of marginalization over increasingly abstract latent variables.

Remark 9.1.3 (The Bayesian BM25 Unit as Recursive Building Block). The unit derived in Section 5 — sigmoid calibration → log-odds aggregation → sigmoid posterior — is a complete single-stage Bayesian inference module. A deep network is a *stack of such modules*, where each module's posterior output becomes the next module's evidence input. This is precisely the recursive structure of hierarchical Bayesian models, where inference proceeds from observed variables through layers of latent variables to the final hypothesis:

$$\begin{aligned} &\underbrace{P(z^{(1)} \mid x)}_{\text{Layer 1: evidence from raw data}} \quad \rightarrow \quad \underbrace{P(z^{(2)} \mid z^{(1)})}_{\text{Layer 2: evidence from evidence}} \\ &\rightarrow \cdots \rightarrow \quad \underbrace{P(y \mid z^{(L)})}_{\text{Output: judgment from constructed evidence}} \end{aligned} \tag{53}$$

The derivation in Section 5 is not limited to shallow inference — it is the *atomic unit* from which arbitrarily deep inference chains are composed.

9.2 Question Sequencing in Architecture Design

The results of Section 6 — sigmoid for belief, ReLU for quantity, softmax for selection — imply that choosing an activation function for a layer is equivalent to choosing the probabilistic question that layer asks. Network architecture design thus becomes **question sequencing**: specifying the order in which probabilistic questions are posed to the data.

Proposition 9.2.1 (Question Sequencing in Standard Architectures). The activation function sequence of major architectures corresponds to a well-defined probabilistic interrogation pipeline:

Architecture	Layer Sequence	Question Sequence
ResNet	ReLU → ... → ReLU → Softmax	"How much feature?" → ... → "How much feature?" → "Which class?"
Transformer (LLM)	GELU → Softmax (attn) → ... → Softmax	"Expected evidence under noise?" → "Which is relevant?" → ... → "Which token?"
Classic MLP	Sigmoid → ... → Sigmoid	"How probable?" → ... → "How probable?"
Sigmoid + Softmax	Sigmoid (hidden) → Softmax	"How probable per feature?" → "Which class?"

Remark 9.2.2 (Why Activation Swaps Change Performance). Empirical studies frequently report that replacing one activation function with another (e.g., ReLU → GELU) changes model performance without a clear theoretical explanation. The question-sequencing framework provides one: the swap changes the **type of question** the layer asks. Replacing ReLU with GELU in hidden layers changes the question from *"how much feature is present?"* (hard thresholding, exact zeros) to *"what is the expected evidence accounting for noise?"* (soft gating, noise-robust). Performance improvements from such swaps correspond to choosing a question better suited to the data distribution — noisy inputs benefit from noise-robust questions.

Remark 9.2.3 (Design Implications). Rather than searching over activation functions empirically, the framework suggests selecting activations by asking: *What type of probabilistic question should each layer pose?* For early layers processing raw, noisy input, GELU or Swish may be appropriate (noise-robust quantity estimation). For intermediate selection layers, softmax attention is natural (context-dependent model selection). For final decision layers, sigmoid or softmax is necessary (posterior probability computation). This transforms architecture design from combinatorial search to principled question selection.

9.3 Reverse Interpretability: Reading the Questions a Network Asks

The forward direction (Section 9.2) uses the framework to *design* networks. The reverse direction uses it to *interpret* existing networks by reading off which probabilistic question each layer is asking.

Proposition 9.3.1 (Activation-Based Interpretability). Given a trained network with known activation functions, the probabilistic question framework (Section 6) assigns a semantic interpretation to each layer without inspecting weights, gradients, or activations:

(i) **Classic sigmoid MLP** (sigmoid hidden → sigmoid output): Every layer asks *"how probable?"* — the network performs shallow iterated Bayesian inference, estimating posterior probabilities at each stage. This explains why sigmoid MLPs behave like stacked logistic regressions.

(ii) **ResNet** (ReLU hidden → softmax output): Hidden layers ask *"how much of each feature is present?"* (sparse quantity detection), and the output asks *"which class?"* (selection). The network performs hierarchical evidence accumulation: sparse features are progressively extracted and then classified. The deep ReLU stack implements a hierarchy of increasingly abstract sparse feature detectors.

(iii) **Transformer** (GELU hidden → softmax attention → softmax output): Hidden GELU layers ask *"what is the expected evidence under noise?"* (noise-robust feature extraction). Attention softmax layers ask *"which features are relevant to the current context?"* (Bayesian model averaging, Section 8). Output softmax asks *"which token/class?"* (final selection). The network performs noise-robust feature extraction, followed by context-dependent evidence selection, followed by posterior judgment.

(iv) **Swish-based networks**: Hidden layers ask *"what is the probability-weighted evidence?"* — passing evidence in proportion to its estimated relevance ($x \cdot \sigma(x)$). This implements uncertainty-aware feature extraction, where the network *simultaneously* estimates quantity and relevance at each layer.

Remark 9.3.2 (Beyond Post-Hoc Inspection). Standard interpretability methods (saliency maps, attention visualization, probing classifiers) inspect the *values* that flow through a network. The question-sequencing framework operates at a different level: it interprets the *type of computation* each layer performs, based solely on its activation function. The two approaches are complementary — one reads the answers, the other reads the questions.

---

10. Discussion

10.1 Reversal of Explanatory Direction

Observation 10.1.1 (Standard Direction). The standard narrative in machine learning proceeds from neural networks to probabilistic interpretation: one *constructs* a neural architecture, then *analyzes* it probabilistically. Bayesian neural networks (Neal, 1996), variational inference (Blundell et al., 2015), and probabilistic deep learning (Gal & Ghahramani, 2016) all follow this direction.

Observation 10.1.2 (Reversed Direction). Our derivation proceeds from probability to neural structure. The neural architecture is not a design decision but a mathematical consequence. This reversal carries significant implications for the theoretical foundations of neural computation.

Theorem 10.1.3 (Generality of Emergence). If the neural structure is a *consequence* of probabilistic inference rather than a *design decision*, then it is not specific to any particular engineering choice. Any system that performs Bayesian calibration of multiple binary evidence signals and combines them through principled log-odds accumulation will inevitably instantiate a feedforward neural computation with sigmoid activations.

10.2 Interpretability by Construction

Proposition 10.2.1 (Constructive Interpretability). Neural networks derived from probabilistic inference are interpretable by construction. Each component has a well-defined role:

Layer	Probabilistic Interpretation
First sigmoid activation	Bayesian calibration: $P(\mathrm{relevant} \mid \mathrm{signal}_i)$
Log-space aggregation	Evidence accumulation: average strength of evidence
Additive bias ($\alpha \log n$)	Agreement bonus: confidence boost from multi-signal concurrence
Second sigmoid activation	Posterior computation: final $P(\mathrm{relevant} \mid \mathrm{all\ signals})$

Remark 10.2.2 (Contrast with Post-Hoc XAI). Modern Explainable AI approaches attempt to understand neural networks by inspecting them *after* training — through activation analysis, attention maps, or gradient-based attribution. These methods treat the network as a black box to be reverse-engineered. In our framework, there is nothing to reverse-engineer: the derivation *is* the explanation.

10.3 A Research Program

Conjecture 10.3.1 (Inverse Derivation Program). For a broader class of neural architectures beyond the two-layer sigmoid network derived here, it may be possible to identify the implicit probabilistic inference problem whose analytical solution produces the given architecture. If successful, this would provide a principled interpretability method: networks would be understood not through post-hoc inspection but through the mathematical structure of the question they answer.

Remark 10.3.2 (Partial Resolution). Sections 6, 8, and 9 provide concrete instances of the inverse derivation program:

Activation / Structure	Probabilistic Question	Status
Sigmoid	Bayesian posterior for binary relevance	Proven (Theorem 6.4.1)
ReLU	MAP estimate under sparse non-negative prior	Proven (Theorem 6.5.3)
Softmax	Categorical exponential family canonical link	Proven (Remark 6.4.3)
Attention	Bayesian model averaging with context-dependent reliability	Proven (Theorem 8.3)
Depth	Recursive marginalization over latent variables	Proven (Theorem 9.1.1)
GELU	Expected activation under Gaussian gating	Conjectured (6.7.1)
Swish	Probability-weighted evidence	Conjectured (6.7.2)

The five proven cases establish that the program is productive: the dominant activation functions, the attention mechanism, and the necessity of depth each correspond to well-defined probabilistic inference problems. The two conjectured cases suggest that the program may be completable for the full standard repertoire.

10.4 Three Traditions, One Structure

Observation 10.4.1 (Convergence). Three intellectual traditions, developed independently across different decades, converge on the same computational structure:

Probability theory asks: given evidence, what is the posterior probability of a hypothesis? The answer involves Bayes' theorem, likelihood ratios, and the logistic function as the canonical link for binary outcomes.

Information retrieval asks: given query terms and document features, how relevant is this document? The answer involves BM25 scoring, IDF weighting, and — as shown in Jeong (2026) and the present paper — sigmoid calibration with log-odds evidence accumulation.

Neural computation asks: given input signals and parameters, what is the output activation? The answer involves weighted linear combination, bias terms, and nonlinear activation functions.

All three produce the same computational graph:

$$\text{inputs} \xrightarrow{\text{linear}} \sigma \xrightarrow{\text{linear}} \sigma \to \text{output} \tag{54}$$

10.5 Robertson's Completed Circle

In 1976, Robertson introduced the Probability Ranking Principle, opening a door between probability theory and information retrieval. BM25 was derived from this probabilistic foundation, yet its scores are not probabilities. For nearly fifty years, this circle remained unclosed.

Bayesian BM25 (Jeong, 2026) closes the circle by returning BM25 scores to the probability space from which the framework originated. The present paper reveals what lies on the other side of the closed door: the computational structure that a separate scientific tradition, developed over different decades and for different purposes, would call a *neural network*.

---

11. Scope and Anticipated Objections

We address several potential objections to sharpen the scope and limitations of our claims.

11.1 Scope of the Isomorphism

Objection. *"The derivation produces only a two-layer sigmoid network. Modern deep learning uses deep architectures with ReLU, GELU, or Transformer blocks. The result is therefore of limited relevance to contemporary neural computation."*

Response. We make four observations. First, the claim is not that all neural networks arise from probabilistic inference, but that at least one concrete architecture does — establishing an existence proof. The scope is deliberately narrow: we derive the structure that arises from a specific, well-defined probabilistic question.

Second, the scope of the derivation extends beyond the sigmoid. Section 6.5 shows that ReLU — the dominant hidden-layer activation — is independently derivable as the MAP estimator under sparse non-negative priors, answering a complementary probabilistic question (*"how much?"* rather than *"how probable?"*). The standard practice of using ReLU in hidden layers and sigmoid at the output (Theorem 6.6.2) thus corresponds to a two-phase probabilistic inference: sparse feature detection followed by posterior estimation. This accounts for the activation functions of the entire feedforward network, not only the output layer.

Third, the distance between the derived architecture and modern attention-based models is shorter than it appears. As shown in Section 8, allowing the aggregation weights to depend on the query-signal interaction — a single conceptual extension — transforms the static feedforward structure into the attention mechanism. The derived network is not a dead end but a starting point from which contemporary architectures are reachable through natural probabilistic generalizations.

Fourth, the derivation is not limited to shallow networks. Theorem 9.1.1 establishes that the derived unit is the atomic building block of arbitrarily deep inference chains: each layer constructs the evidence required by the next through iterated marginalization over latent variables. Depth arises because high-level judgments require evidence that does not exist in the raw data and must be progressively constructed (Section 9.1).

11.2 The Independence Assumption

Objection. *"The Bayesian derivation assumes independence among scoring signals. In practice, text and vector scores are correlated. The independence assumption is unrealistic."*

Response. The independence assumption determines the *initial structure* — the architecture and its starting parameters — not the final operating state. In the neural network interpretation, the uniform weights $w_i = 1/n$ (Theorem 5.3.1) correspond to the Naive Bayes starting point. When these weights become learnable parameters (Remark 5.3.2), the network can capture signal dependencies through training. Our derivation thus provides a principled initialization: the network begins at the Naive Bayes solution and refines toward the true posterior through weight adaptation. This is analogous to how Xavier or He initialization provides a theoretically motivated starting point for gradient-based learning.

11.3 The Conjunction Bonus

Objection. *"The bias term $\alpha \log n$ is a fixed constant, whereas neural network biases are learned parameters. This weakens the correspondence."*

Response. The value $\alpha \log n$ is not arbitrary. It encodes the statistical principle that n independent agreeing observations increase confidence by $\sqrt{n}$ (Theorem 4.4.1). In the neural network interpretation, this provides a theoretically grounded inductive bias for the output neuron's bias parameter. Rather than initializing the bias at zero — the standard practice with no theoretical justification — one could initialize at $\alpha \log n$, encoding the prior knowledge that multi-signal agreement constitutes evidence. Whether this initialization improves convergence or generalization is an empirical question, but its theoretical motivation is clear.

11.4 The Asymptotic Linearity Condition

Objection. *"The neuron isomorphism requires $\bar{P} \ll 1$ for exact correspondence in the log-probability domain (Theorem 5.2.1). Is this a limiting assumption?"*

Response. This condition is the natural operating regime of information retrieval. In a corpus of millions of documents, the fraction relevant to any given query is vanishingly small. The rare relevance assumption ($\bar{P} \ll 1$) is not a limitation but a description of the domain. Moreover, the isomorphism is exact in the log-odds domain regardless of $\bar{P}$ (Theorem 5.2.1, exact correspondence), so the asymptotic condition is needed only for the specific parametrization using log-probabilities as neuron inputs.

---

12. Related Work

12.1 Bayesian Approaches to Neural Networks

Neal (1996) established the connection between infinitely wide neural networks and Gaussian processes, providing a Bayesian interpretation of neural network priors. Blundell et al. (2015) introduced practical weight uncertainty methods. Gal and Ghahramani (2016) showed that dropout can be interpreted as approximate Bayesian inference. All of these proceed in the direction from neural networks to probability — the reverse of our derivation.

12.2 Probabilistic Foundations of Information Retrieval

The probabilistic relevance framework (Robertson & Zaragoza, 2009) provides the theoretical foundation for BM25. Lafferty and Zhai (2001) developed language modeling approaches with probabilistic foundations. Metzler and Croft (2005) introduced Markov random field models for IR. Our work completes the probabilistic program initiated by Robertson by returning BM25 to probability space and revealing the neural structure implicit in multi-signal probabilistic retrieval.

12.3 Score Calibration and Fusion

Platt (1999) introduced sigmoid calibration for SVM outputs, establishing the empirical effectiveness of sigmoid transformations for probability calibration. Our derivation provides a Bayesian justification for this approach, showing that the sigmoid is not merely an effective calibrator but the unique function satisfying the constraints of binary probabilistic inference.

12.4 Neural Network Pruning

Structured and unstructured pruning methods (Han et al., 2015; Li et al., 2017) reduce neural network inference cost through approximate elimination of parameters. Our observation that WAND/BMW constitute *exact* pruning methods (Section 7) contrasts with these approximate approaches and depends on the bounded, monotonic properties of the sigmoid activation — properties that follow from the probabilistic derivation.

12.5 Sparse Coding and Activation Functions

Olshausen and Field (1997) demonstrated that sparse coding with non-negative constraints produces receptive fields resembling biological neurons, establishing the connection between sparsity priors and neural activation patterns. Nair and Hinton (2010) introduced ReLU as a practical activation function, observing its connection to sparse representations without providing a formal probabilistic derivation. Glorot et al. (2011) analyzed the sparsity properties of ReLU empirically. Our derivation in Section 6.5 completes this line of work by formally proving that ReLU is the unique MAP estimator under sparse non-negative priors — providing the probabilistic foundation that was previously only empirically observed. Hendrycks and Gimpel (2016) introduced GELU and Ramachandran et al. (2018) proposed Swish, both through empirical search over activation function spaces. Our conjectures (Section 6.7) suggest that these too may admit probabilistic derivations, potentially unifying the proliferation of activation functions under a single theoretical framework.

---

13. Conclusion

We set out to answer a question in information retrieval: *what is the probability that a document is relevant given multiple evidence signals?* We applied Bayes' theorem and the sigmoid emerged as the posterior (Section 2.3). We resolved the conjunction shrinkage problem through geometric mean aggregation and log-odds combination, and a second sigmoid emerged as the final posterior computation (Section 4). We examined the end-to-end structure and recognized a two-layer feedforward neural network (Section 5).

The neuron was not designed. It was *derived* — latent within the structure of probabilistic inference over binary relevance judgments. This analytical emergence establishes several results:

1. **Neural structure as theorem**: At least one concrete neural architecture — the two-layer sigmoid network — arises as a mathematical consequence of Bayesian inference, not as an engineering design. Moreover, the derived structure is not an isolated artifact: relaxing the uniform reliability assumption extends it to the attention mechanism (Section 8), revealing a continuous path from first-principles probability to the building blocks of modern Transformers. The derivation provides an existence proof that neural architectures can be theorems of probability.
2. **Activation functions as probabilistic answers**: The sigmoid and ReLU — the two dominant activation functions in deep learning — are derived from complementary probabilistic questions. Sigmoid is the canonical link for binary belief (*"how probable?"*), and ReLU is the MAP estimator for sparse feature presence (*"how much?"*). Their complementarity explains the standard practice of using ReLU in hidden layers and sigmoid at the output: feature detection followed by probability estimation. This framework extends to softmax (multi-class canonical link), with GELU and Swish as conjectured instances.
3. **Depth as recursive Bayesian inference**: Deep networks are chains of iterated marginalization over latent variables (Theorem 9.1.1). Each layer constructs the evidence required by the next — from raw pixels to edges to shapes to classes — because the evidence needed for high-level judgments does not exist in the raw data. The derived inference unit is not limited to shallow models; it is the atomic building block from which arbitrarily deep inference chains are composed (Section 9.1).
4. **Attention as Bayesian model averaging**: The weighted sum at the core of attention is not an arbitrary aggregation choice but the probabilistically correct method for combining uncertain evidence in log-space. Our framework answers a question that the original attention formulation left open: *why* a weighted sum, and not some other aggregation? The answer is that Bayesian evidence combination in the log-odds domain is inherently additive.
5. **Interpretability by construction and by question**: Each layer of the derived network corresponds to a well-defined step in Bayesian inference (Section 10.2). Beyond the derived network, the question-sequencing framework (Section 9.2) enables interpretability of *arbitrary* architectures: each activation function identifies the probabilistic question its

layer asks, providing a semantic reading of network structure without inspecting weights or gradients (Section 9.3).

6. **Exact pruning from IR**: The bounded, monotonic properties of the sigmoid — inherited from its probabilistic origin — enable exact WAND/BMW pruning with formal safety guarantees. The unbounded properties of ReLU — equally inherited from its probabilistic origin — provide complementary structural sparsity. Together, they suggest a neural inference architecture mirroring the IR pipeline: inverted index construction (ReLU sparsity) followed by query-time safe pruning (sigmoid boundedness).
7. **Inevitability of the sigmoid**: The sigmoid's recurrence across Bayesian inference, information retrieval, and neural computation is a consequence of the exponential family structure of Bernoulli random variables. Any system processing binary evidence under the natural constraints of probability will arrive at the same function.

The mathematics does not care what we call things. Whether we say "Bayesian posterior" or "sigmoid neuron," "sparse feature detector" or "ReLU unit," "evidence accumulation" or "attention" — the same mathematical structures appear wherever information is processed under uncertainty. The neuron is not an invention of neuroscience or machine learning. It is a theorem of probability.

References

1. Bengio, Y., Léonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.
2. Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. *Proceedings of the 32nd International Conference on Machine Learning*, 1613–1622.
3. Broder, A. Z., Carmel, D., Herscovici, M., Soffer, A., & Zien, J. (2003). Efficient query evaluation using a two-level retrieval process. *Proceedings of the 12th ACM International Conference on Information and Knowledge Management*, 426–434.
4. Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search Engines: Information Retrieval in Practice*. Addison-Wesley.
5. Ding, S., & Suel, T. (2011). Faster top-k document retrieval using block-max indexes. *Proceedings of the 34th International ACM SIGIR Conference*, 993–1002.
6. Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of the 33rd International Conference on Machine Learning*, 1050–1059.
7. Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 315–323.
8. Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. *Advances in Neural Information Processing Systems*, 28.
9. Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*.
10. Jeong, J. (2026). Bayesian BM25: A probabilistic framework for hybrid text and vector search. *Cognica Technical Report*. DOI: 10.5281/zenodo.18414941.

11. Kurtz, M., Kopinsky, J., Gelber, A., Naumov, M., Diamos, G., & Kolkin, N. (2020). Inducing and exploiting activation sparsity for fast inference on deep neural networks. *Proceedings of the 37th International Conference on Machine Learning*.
12. Lafferty, J., & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. *Proceedings of the 24th Annual International ACM SIGIR Conference*, 111–119.
13. Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2017). Pruning filters for efficient convnets. *International Conference on Learning Representations*.
14. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
15. Metzler, D., & Croft, W. B. (2005). A Markov random field model for term dependencies. *Proceedings of the 28th Annual International ACM SIGIR Conference*, 472–479.
16. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning*, 807–814.
17. Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer.
18. Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23), 3311–3325.
19. Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3), 61–74.
20. Ramachandran, P., Zoph, B., & Le, Q. V. (2018). Searching for activation functions. *International Conference on Learning Representations (Workshop)*.
21. Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation*, 33(4), 294–304.
22. Robertson, S. E., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389.
23. Teerapittayanon, S., McDanel, B., & Kung, H. T. (2016). BranchyNet: Fast inference via early exiting from deep neural networks. *Proceedings of the 23rd International Conference on Pattern Recognition*, 2464–2469.
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.