

Article

A Convergent Method for Energy Optimization in Modern Hopfield Networks

Yida Bao ¹, Mohammad Arifuzzaman ^{1,*}, Tran Duc Le ¹, Tao Jiang ², Jing Hou ³, Yuan Xing ⁴
and Dongfang Hou ⁴

¹ Mathematics, Statistics & Computer Science Department, University of Wisconsin Stout, Menomonie, WI 54751, USA; baoy@uwstout.edu (Y.B.); let@uwstout.edu (T.D.L.)

² Mid-Florida Research and Education Center (MREC), University of Florida, Apopka, FL 32703, USA; taojiang@ufl.edu

³ Department of Computer Science and Engineering, California State University San Marcos, San Marcos, CA 92096, USA; jhou@csusm.edu

⁴ Engineering & Technology Department, University of Wisconsin Stout, Menomonie, WI 54751, USA; xingy@uwstout.edu (Y.X.); houd@uwstout.edu (D.H.)

* Correspondence: arifuzzamanm@uwstout.edu

Abstract

Modern Hopfield networks are energy-based associative memory models whose performance critically depends on the structure and optimization of their energy functions. While recent formulations substantially improve storage capacity, the resulting non-convex energy landscapes are often optimized using heuristic update rules that can be sensitive to initialization and may not provide monotonic energy descent or rigorous convergence guarantees. In this work, we propose a new energy formulation for modern Hopfield networks together with a principled iterative optimization scheme. The proposed energy admits a natural decomposition that allows optimization via the concave–convex procedure (CCCP), yielding well-defined network dynamics with guaranteed energy descent beyond classical Hopfield updates. We establish fundamental theoretical properties of the proposed framework, including non-negativity, boundedness, and monotonic decrease in the energy along iterations. In particular, we prove that the induced dynamics converge to a stationary point of the energy function, providing explicit convergence guarantees for the resulting Hopfield-type model. We further evaluate the proposed approach on synthetic classification tasks and compare its optimization behavior with that of the original Hopfield network and several standard machine learning baselines. Experimental results demonstrate improved stability, convergence behavior, and competitive classification performance. We also validate the approach on real-world benchmark datasets to demonstrate utility beyond controlled experiments. Overall, this work provides a theoretically grounded energy-based optimization framework for modern Hopfield networks, clarifying the role of principled optimization in achieving stable and convergent associative memory dynamics.

Keywords: modern Hopfield networks; energy-based models; energy function optimization; convergence analysis; associative memory; non-convex optimization; concave–convex procedure; theoretical guarantees



Academic Editors: María del Carmen Pegalajar Jiménez and Luis G. Baca Ruiz

Received: 25 December 2025

Revised: 16 February 2026

Accepted: 26 February 2026

Published: 28 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Hopfield-type energy-based neural models have recently regained broad attention in machine learning. Associative memory networks, epitomized by the Hopfield network,

store patterns as attractors in a dynamical system. In a Hopfield network, neurons update their states to minimize an energy function, making the network a content-addressable memory that converges to the stored pattern closest to an initial cue. Crucially, the form of the energy function governs the network's dynamics and convergence properties [1]. In the classical Hopfield model [2,3], the energy is a quadratic form (essentially a Hebbian weight sum), which serves as a Lyapunov function: with symmetric connections and asynchronous updates, the energy monotonically decreases and the network is guaranteed to converge to a stable fixed point (a local minimum of the energy) [3]. This guarantees that retrieval will end at some attractor state, which in an ideal case corresponds to a stored pattern [2,3]. However, the quadratic energy landscape also limits storage capacity and can admit spurious minima [3]. For random patterns, a Hopfield network of N binary neurons can robustly store only on the order of 0.14, N patterns (approximately 14% of its neurons) before spurious memories and errors dominate [4,5]. Here 0.14 refers to the classical critical loading ratio $\alpha_c \approx 0.14$ obtained from statistical-physics analyses of the Hopfield model with random i.i.d. patterns [4,5]. Let P denote the number of stored patterns. Classical analyses show that reliable retrieval breaks down when the ratio P/N exceeds a critical value (approximately 0.14 for i.i.d. random patterns), because cross-talk from non-target patterns increases and destabilizes the desired fixed points, leading to spurious attractors and errors [2,4,5]. This capacity limitation, along with the presence of undesired local minima, motivated research into modified energy functions that could improve memory storage and retrieval capabilities [6,7].

Modern Hopfield networks (also known as dense associative memories) address the above limitations by redesigning the energy function [1,6]. A line of work has shown that altering the energy's functional form can dramatically increase the number of stably stored patterns. Krotov [6] and Hopfield introduced a higher-order energy function using a polynomial interaction term in place of the classical quadratic term, significantly boosting storage capacity beyond the Hopfield 0.14, N limit. Demircigil et al. [8] further extended this idea by employing an exponential interaction (effectively a smooth maximum or softmax over pattern matches) in the energy; in this limit case, the network can achieve exponential capacity in N (storing on the order of $2^{N/2}$ patterns). More recently, Ramsauer et al. [1] proposed a modern continuous Hopfield network in which the state is a continuous vector and the energy function is based on a softmax-type aggregation of pattern similarities. Their update rule was shown to be mathematically equivalent to the softmax attention mechanism from Transformers, retrieving a stored pattern in a single step by an attention-weighted sum [1]. These modern Hopfield networks, with softmax-based energies, preserve the convergence principle (energy is still a Lyapunov function) but operate on a different, highly non-quadratic energy landscape [2]. The softmax-based energy function dramatically alters the landscape's shape for example, creating a dominant global basin for the most similar pattern rather than many shallow basins and thus requires careful theoretical analysis to ensure the dynamics still converge [8].

In this paper, we introduce a novel energy formulation for a modern Hopfield network based on a softmin-type aggregation mechanism [9]. It induces a qualitatively different energy landscape, giving rise to distinct fixed points and convergence behavior that require independent theoretical analysis. A fixed point is a state ζ^* that remains unchanged under the network update, i.e., $\zeta^* = \mathcal{F}(\zeta^*)$. We prove that the proposed Hopfield energy remains non-negative and bounded [9], and we show that under the associated update rule the energy decreases monotonically at each iteration, guaranteeing convergence to a local minimum. This convergence establishes the associative memory property of the proposed softmin-based Hopfield network. Furthermore, we provide an interpretation of the resulting optimization dynamics through the Concave–Convex Procedure (CCCP) [10],

a general framework for optimizing non-convex functions. In our context, CCCP offers a principled way to derive the network update equations by splitting the new energy into concave and convex parts, and it naturally ensures energy descent at each step. We emphasize that this use of CCCP is as an analytical tool to understand our model's dynamics rather than a new algorithmic contribution—the main novelty lies in the resulting energy landscape and its associated optimization dynamics.

From a practical perspective, our contribution provides a stable and interpretable optimization principle for Hopfield-type modules embedded in modern learning systems. This is particularly relevant when Hopfield dynamics are used as iterative refinement blocks in tabular prediction, denoising, or retrieval under partial or noisy cues, where non-heuristic convergence behavior is desirable.

2. Related Work

The Hopfield network [2,3] is a seminal model of associative memory in which patterns are stored as attractors of a dynamical system. In the binary Hopfield network, each neuron's state is ± 1 and the symmetric weight matrix is typically set by Hebbian learning to encode the patterns. Hopfield showed that such networks possess a Lyapunov energy function: for appropriate symmetric weights (and no self-loops), asynchronous neuron updates provably decrease the energy until a stable state is reached. In other words, every update step brings the network closer to a local minimum of the energy function, guaranteeing convergence to a fixed-point memory. This elegant energy-based convergence result sparked extensive analysis of Hopfield networks in the 1980s [2,3]. Amit et al. [11] and others characterized these capacity limits and the nature of spurious states in the quadratic energy landscape. Variants of the Hopfield model, including networks with biased neurons or unit normalization, were also studied for their convergence and stability properties, all within the same general energy-minimization framework.

In recent years, modern Hopfield networks, also referred to as dense associative memories, have attracted renewed attention due to their ability to overcome the severe storage limitations of the classical Hopfield model. Unlike the original formulation, which relies on a quadratic energy function, these modern variants employ more expressive, non-linear aggregation mechanisms over stored patterns, leading to substantially richer energy landscapes. Krotov and Hopfield [6] initiated this line of work by proposing an energy function of the form

$$E(\xi) = - \sum_i F(x_i^\top \xi),$$

where $\{x_i\}$ denote the stored patterns and $F(\cdot)$ is a higher-order interaction function. By choosing F as a power-law (polynomial) function, they demonstrated that the network can store a super-linear, polynomial-in- N number of memories, significantly exceeding the classical Hopfield capacity. Building on this idea, Demircigil et al. [8] considered the limit of infinite polynomial order by adopting an exponential interaction function, $F(z) = \exp(z)$, which leads to the energy

$$E(\xi) = - \sum_i \exp(x_i^\top \xi).$$

This formulation is closely related to a softmax operation over stored patterns in the computation of the energy minimum and was shown to achieve exponential storage capacity in the system dimension N , albeit with increasingly narrow basins of attraction as the number of stored patterns grows [8].

Despite these fundamental changes in capacity and energy geometry, discrete-state modern Hopfield networks retain convergence guarantees analogous to the classical model [8]. The associated energy function serves as a Lyapunov function under the net-

work update dynamics, ensuring convergence to a local minimum. However, the resulting energy landscape is markedly sharper, characterized by deep minima at stored patterns and strong penalties for states that fail to align with any memory [1].

Building on these developments, Ramsauer et al. [1] introduced a continuous-state formulation of the modern Hopfield network, whose update rule is directly inspired by, and mathematically equivalent to the softmax attention mechanism used in Transformer architectures [12]. In this model, the network state $\xi \in \mathbb{R}^d$ is continuous, and the associated energy function admits a log-sum-exp representation, supplemented with regularization terms to ensure boundedness of $\|\xi\|$. Minimization of this energy leads to the fixed-point update

$$\xi^{\text{new}} = X \text{softmax}(\beta X^\top \xi),$$

where $X = [x_1, \dots, x_N]$ collects the stored patterns and β controls the sharpness of the interaction. This update can be interpreted as computing the minimizer of a log-sum-exp-type energy function under the corresponding update dynamics. Under this update, the new state is obtained as a softmax-weighted average of all stored patterns, effectively retrieving the pattern with the highest similarity to the current state [1]. This mechanism explains the characteristic one-shot or few-step convergence behavior observed in modern Hopfield networks. Despite the non-convexity of the underlying energy function, Ramsauer et al. [1] established rigorous convergence guarantees for the continuous Hopfield dynamics. By interpreting the update rule as a specific energy-decreasing map, they showed that the energy decreases monotonically along the iterates and that all limit points of the dynamics correspond to local minima of the underlying non-quadratic energy function [1]. Consequently, the classical principle of energy-based convergence is preserved as the model transitions from a quadratic to a softmax-defined energy, although the guarantees are inherently local and depend on the precise form of the update rule [1].

The above developments are part of a broader context of energy-based learning and optimization in neural networks [2,13]. Hopfield networks, Boltzmann machines, and related models all define a scalar energy landscape over network states, and network dynamics perform (explicitly or implicitly) a form of gradient descent on that landscape [14,15]. A longstanding challenge with such models is handling their non-convex energy functions—which inherently possess multiple local minima [7]. Convergence results for Hopfield networks guarantee reaching a local minimum, but not a global minimum, and much work has gone into characterizing and improving the landscape of these energy functions [1,16].

One general approach to optimizing non-convex energies is the Concave–Convex Procedure (CCCP) proposed by Yuille and Rangarajan [10]. CCCP is a meta-algorithm that expresses an objective $E(\xi)$ as a sum of a convex part and a concave part,

$$E(\xi) = E_{\text{convex}}(\xi) + E_{\text{concave}}(\xi),$$

and then optimizes it by iteratively tangent-approximating the concave part with a linear term [10]. Yuille and Rangarajan proved that CCCP produces a discrete-time dynamical system that guarantees monotonic decrease in the overall energy at each iteration [10]. Many existing algorithms can be interpreted in terms of CCCP—for example, the Expectation–Maximization (EM) algorithm for mixture models is a special case of CCCP, as are certain variational inference schemes [10,17,18].

In the context of modern Hopfield networks, CCCP provides a useful theoretical interpretation of the update dynamics [1,10]. Indeed, the concave–convex procedure (CCCP) provides a unifying framework for analyzing the update dynamics of modern Hopfield networks, including the continuous formulation studied by Ramsauer et al. By splitting the energy (e.g., into a convex quadratic term and a concave $\log \sum \exp$ term), one

obtains an iterative update that is identical to the network's attention-like update [1,10]. This not only offers an alternative proof that the energy decreases every update, but also places Hopfield retrieval in the same framework as other convergent iterative optimization algorithms [1,2,10]. Our work leverages these insights from CCCP in analyzing the softmin-based energy: we show how the new energy can likewise be decomposed and how the CCCP perspective yields the correct update rule and convergence guarantee.

3. Concave–Convex Procedure Energy Formulation

We will present the core energy formulation underlying the proposed softmin-based Hopfield network. Our objective is to construct a continuous, non-convex energy function whose minimization yields a well-defined network update rule, while admitting rigorous analysis of boundedness, monotonic descent, and convergence. The resulting formulation serves as the theoretical foundation for the optimization dynamics and convergence guarantees developed in the subsequent subsections.

3.1. Energy Function Definition

We define a continuous Hopfield-style energy function for the network state $\xi \in \mathbb{R}^d$ as follows: Let $X = [x_1, \dots, x_N] \in \mathbb{R}^{d \times N}$ denote the matrix of stored patterns, where each $x_i \in \mathbb{R}^d$ satisfies $\|x_i\| \leq M$, and let $\xi \in \mathbb{R}^d$ be the continuous network state. For a given sharpness parameter $\beta > 0$, we define the proposed softmin-based energy function as

$$E(\xi) = \frac{1}{2} \|\xi\|^2 - \frac{1}{\beta} \log \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right) + \frac{1}{\beta} \log N + \frac{1}{2} M^2. \quad (1)$$

The quadratic term $\frac{1}{2} \|\xi\|^2$ acts as a regularizer that penalizes large state magnitudes, while the log-sum-exponential term provides a smooth approximation to the minimum alignment between ξ and the stored patterns. The additive constants $\frac{1}{\beta} \log N$ and $\frac{1}{2} M^2$ are independent of ξ and are included for normalization purposes; they do not affect the minimizers of $E(\xi)$.

Due to the presence of the softmin aggregation term in (1), the energy function $E(\xi)$ is, in general, non-convex. Consequently, direct minimization requires an optimization strategy capable of handling non-convex objectives while preserving descent guarantees.

3.2. Concave–Convex Decomposition

To facilitate principled optimization of the non-convex energy (1), we express $E(\xi)$ as the sum of a convex and a concave component. Specifically, we write

$$E(\xi) = E_{\text{vex}}(\xi) + E_{\text{cave}}(\xi), \quad (2)$$

where

$$E_{\text{vex}}(\xi) = \frac{1}{2} \|\xi\|^2 + \frac{1}{\beta} \log N + \frac{1}{2} M^2, \quad (3)$$

and

$$E_{\text{cave}}(\xi) = -\frac{1}{\beta} \log \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right). \quad (4)$$

The function $E_{\text{vex}}(\xi)$ is convex, as it consists of a quadratic term in ξ plus constants. The function $E_{\text{cave}}(\xi)$ is concave because the log-sum-exp function is convex, and multiplication by $-1/\beta$ preserves concavity. Therefore, the decomposition in (2) constitutes a valid convex–concave decomposition of the energy function.

This decomposition permits the application of the Concave–Convex Procedure (CCCP) [10], which optimizes objectives expressed as the sum of a convex and a con-

cave term. Importantly, CCCP is not introduced here as a new algorithmic contribution, but rather as a principled optimization framework that naturally aligns with the structure of the proposed softmin-based energy. In the subsequent subsection, this decomposition is used to derive the network update dynamics and establish monotonic energy descent.

3.3. Energy Function and Update Rule

We will derive the network update dynamics induced by the proposed energy function. By applying the Concave–Convex Procedure (CCCP) to the convex–concave decomposition established earlier, we obtain a closed-form iterative update rule [10]. This derivation makes explicit the connection between energy minimization and the resulting softmin-based Hopfield dynamics.

The Concave–Convex Procedure (CCCP) applied to E is

$$\nabla E_1(\xi^{t+1}) = -\nabla E_2(\xi^t). \quad (5)$$

which is

$$\nabla_{\xi} \left(\frac{1}{2} \xi^T \xi + C_1 \right) = -\nabla_{\xi} \left(-\frac{1}{\beta} \ln \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right) \right). \quad (6)$$

The left-hand side is equal to ξ^{t+1} .

Using the chain rule, we have:

$$\text{right-hand side} = \frac{1}{\beta} \frac{\partial}{\partial \xi} \ln \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right). \quad (7)$$

With respect to ξ , we apply the chain rule and the properties of logarithmic derivatives. First, the derivative of the natural logarithm is:

$$\frac{d \ln(u)}{du} = \frac{1}{u}. \quad (8)$$

For the inner function:

$$f(\xi) = \sum_{i=1}^N \exp(-\beta x_i^T \xi), \quad (9)$$

its gradient is:

$$\nabla f(\xi) = \sum_{i=1}^N (-\beta x_i) \exp(-\beta x_i^T \xi). \quad (10)$$

Applying the chain rule for differentiation:

$$\begin{aligned} \frac{\partial \ln(f(\xi))}{\partial \xi} &= \frac{1}{f(\xi)} \nabla f(\xi) \\ &= \frac{1}{\sum_{i=1}^N \exp(-\beta x_i^T \xi)} \sum_{i=1}^N (-\beta x_i) \exp(-\beta x_i^T \xi). \end{aligned} \quad (11)$$

Finally, multiplying by $-\frac{1}{\beta}$ (since $E_2(\xi) = -\frac{1}{\beta} \ln(f(\xi))$), we obtain:

$$\begin{aligned} \frac{\partial E_2(\xi)}{\partial \xi} &= -\frac{1}{\beta} \frac{\partial \ln(f(\xi))}{\partial \xi} \\ &= \frac{1}{\sum_{i=1}^N \exp(-\beta x_i^T \xi)} \sum_{i=1}^N x_i \exp(-\beta x_i^T \xi). \end{aligned} \quad (12)$$

Thus, the partial derivative of $E_2(\xi)$ with respect to ξ is:

$$\frac{\partial E_2(\xi)}{\partial \xi} = \frac{1}{\sum_{i=1}^N \exp(-\beta x_i^T \xi)} \sum_{i=1}^N x_i \exp(-\beta x_i^T \xi). \quad (13)$$

The matrix $\mathbf{X}$ is composed of column vectors x_i , i.e.,

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_N \\ | & | & & | \end{bmatrix}, \quad (14)$$

where each x_i is a $d \times 1$ column vector. Thus, $\mathbf{X}$ is of size $d \times N$.

Define the softmax operator by

$$\text{Softmin}(z)_i = \frac{\exp(-z_i)}{\sum_{j=1}^N \exp(-z_j)}. \quad (15)$$

Using (15), we can rewrite the gradient as a matrix-vector multiplication:

$$\frac{\partial E_2(\xi)}{\partial \xi} = \mathbf{X} \text{Softmin}(\beta \mathbf{X}^T \xi). \quad (16)$$

Thus, we can write:

$$\nabla E_2(\xi^t) = \mathbf{X} \text{Softmin}(\beta \mathbf{X}^T \xi^t). \quad (17)$$

Substituting (17) into (5), we obtain the CCCP update rule:

$$\xi^{t+1} = -\mathbf{X} \text{Softmin}(\beta \mathbf{X}^T \xi^t). \quad (18)$$

3.4. Theoretical Guarantees

We establish several fundamental properties of the proposed energy function (1) and the CCCP-induced update rule (18). These results characterize the energy landscape and the optimization dynamics, including (i) non-negativity, (ii) upper boundedness, (iii) monotonic energy decrease under CCCP iterations, and (iv) convergence of the resulting discrete-time dynamical system.

Theorem 1 (Non-negativity). *For the energy function $E(\xi)$ defined in (1), we have*

$$E(\xi) \geq 0 \quad \text{for all } \xi \in \mathbb{R}^d. \quad (19)$$

Proof of Theorem 1. We recall the energy function

$$E = \frac{1}{2} \xi^T \xi - \frac{1}{\beta} \ln \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right) + \frac{1}{\beta} \ln(N) + \frac{1}{2} M^2. \quad (20)$$

The matrix $\mathbf{X}$ is composed of column vectors x_i , and we define

$$M = \max_i \|x_i\|. \quad (21)$$

For a given state ξ , define

$$x_{\xi} = x_k \quad \text{where} \quad k = \arg \min_i (\xi^T x_i). \quad (22)$$

We rewrite the energy as

$$\begin{aligned} E &= -\frac{1}{\beta} \ln \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right) + \frac{1}{\beta} \ln(N) + \frac{1}{2} \xi^T \xi + \frac{1}{2} M^2 \\ &= -\frac{1}{\beta} \ln \left(\frac{1}{N} \sum_{i=1}^N \exp(-\beta x_i^T \xi) \right) + \frac{1}{2} \xi^T \xi + \frac{1}{2} M^2. \end{aligned} \quad (23)$$

Since the logarithm is monotone increasing and the average of nonnegative terms is lower bounded by any single term, we have

$$\frac{1}{N} \sum_{i=1}^N \exp(-\beta x_i^T \xi) \geq \frac{1}{N} \exp(-\beta x_{\xi}^T \xi). \quad (24)$$

Taking logarithms yields

$$\ln \left(\frac{1}{N} \sum_{i=1}^N \exp(-\beta x_i^T \xi) \right) \geq -\beta x_{\xi}^T \xi - \ln N. \quad (25)$$

Substituting this inequality into (23), we obtain

$$\begin{aligned} E &\geq -\frac{1}{\beta} \ln \left(\exp(-\beta x_{\xi}^T \xi) \right) + \frac{1}{2} \xi^T \xi + \frac{1}{2} M^2 \\ &= x_{\xi}^T \xi + \frac{1}{2} \xi^T \xi + \frac{1}{2} M^2. \end{aligned} \quad (26)$$

Since $\|x_{\xi}\| \leq M$, we further have

$$x_{\xi}^T x_{\xi} \leq M^2, \quad (27)$$

and therefore

$$\begin{aligned} E &\geq -\frac{1}{\beta} \left(-\beta x_{\xi}^T \xi - \ln N \right) + \frac{1}{2} \xi^T \xi + \frac{1}{2} M^2 \\ &= x_{\xi}^T \xi + \frac{1}{\beta} \ln N + \frac{1}{2} \xi^T \xi + \frac{1}{2} M^2. \end{aligned} \quad (28)$$

The final expression is a squared Euclidean norm, which completes the proof. $\square$

Theorem 2 (Upper boundedness). *For the energy function $E(\xi)$ defined in (1), there exists a finite constant $C < \infty$ such that*

$$E(\xi) \leq C \quad \text{for all } \xi \in \mathbb{R}^d \text{ with } \|\xi\| \leq M. \quad (29)$$

Proof of Theorem 2. Consider the energy function

$$E = -\frac{1}{\beta} \ln \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right) + \frac{1}{\beta} \ln(N) + \frac{1}{2} \xi^T \xi + \frac{1}{2} M^2. \quad (30)$$

The logarithmic term is related to the log-sum-exp function. Since the function

$$f(u) = -\ln(u) \quad (31)$$

is convex for $u > 0$, Jensen's inequality applies to the random variables

$$Z_i = \exp(-\beta x_i^T \xi), \quad (32)$$

with uniform weights $a_i = \frac{1}{N}$.

Applying Jensen's inequality yields

$$-\ln\left(\frac{1}{N}\sum_{i=1}^N Z_i\right) \leq -\frac{1}{N}\sum_{i=1}^N \ln(Z_i). \quad (33)$$

Substituting $Z_i = \exp(-\beta x_i^T \xi)$ and using $\ln(\exp(y)) = y$, we obtain

$$-\ln\left(\frac{1}{N}\sum_{i=1}^N \exp(-\beta x_i^T \xi)\right) \leq \frac{\beta}{N}\sum_{i=1}^N x_i^T \xi. \quad (34)$$

Substituting this bound into the expression for E , we have

$$E \leq \frac{1}{N}\sum_{i=1}^N x_i^T \xi + \frac{1}{2}\xi^T \xi + \frac{1}{2}M^2. \quad (35)$$

Define the mean pattern

$$x_{\text{mean}} = \frac{1}{N}\sum_{i=1}^N x_i. \quad (36)$$

Then (35) can be written as

$$E \leq x_{\text{mean}}^T \xi + \frac{1}{2}\xi^T \xi + \frac{1}{2}M^2. \quad (37)$$

Since $\|x_{\text{mean}}\| \leq M$, by the Cauchy–Schwarz inequality we have

$$x_{\text{mean}}^T \xi \leq \|x_{\text{mean}}\| \|\xi\| \leq M\|\xi\|. \quad (38)$$

Applying the inequality $ab \leq \frac{1}{2}(a^2 + b^2)$ yields

$$x_{\text{mean}}^T \xi \leq \frac{1}{2}\|\xi\|^2 + \frac{1}{2}M^2. \quad (39)$$

Therefore,

$$\begin{aligned} E &\leq \frac{1}{2}\|\xi\|^2 + \frac{1}{2}M^2 + \frac{1}{2}\|\xi\|^2 + \frac{1}{2}M^2 \\ &= \|\xi\|^2 + M^2. \end{aligned} \quad (40)$$

Under the condition $\|\xi\| \leq M$, we have $\|\xi\|^2 \leq M^2$, and hence

$$E(\xi) \leq \|\xi\|^2 + M^2 \leq 2M^2. \quad (41)$$

Taking $C = 2M^2$ completes the proof. $\square$

In our network dynamics, the state ξ is constrained to a bounded set determined by the stored patterns, which implies $\|\xi\| \leq M$ along the trajectory. Therefore, Theorem 2 provides a uniform upper bound on the energy along the network updates.

Theorem 3 (Monotonic energy decrease under CCCP). *Let $\{\xi^{(t)}\}_{t \geq 0}$ be generated by the CCCP update associated with the energy*

$$E(\xi) = E_1(\xi) + E_2(\xi), \quad (42)$$

where

$$E_1(\xi) = \frac{1}{2}\xi^T \xi + \frac{1}{\beta} \ln(N) + \frac{1}{2}M^2 = \frac{1}{2}\xi^T \xi + C_1, \quad (43)$$

and

$$E_2(\xi) = -\frac{1}{\beta} \ln \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right). \quad (44)$$

Then the energy is monotonically non-increasing along the iterations:

$$E(\xi^{(t+1)}) \leq E(\xi^{(t)}) \quad \text{for all } t \geq 0. \quad (45)$$

Proof of Theorem 3. Consider the CCCP (Concave–Convex Procedure) algorithm [10]. CCCP is a majorization–minimization algorithm used to solve optimization problems of the form:

$$\min f(x) \quad \text{where} \quad f(x) = u(x) - v(x), \quad (46)$$

where u and v are convex functions [19,20]. The CCCP algorithm is an iterative process that solves the following sequence of convex programs:

$$x^{(t+1)} \in \arg \min \left(u(x) - x^T \nabla v(x^{(t)}) \right), \quad (47)$$

where $\nabla v(x^{(t)})$ denotes the gradient of $v(x)$ at $x^{(t)}$ [21].

We already have the energy function

$$E(\xi) = E_1(\xi) + E_2(\xi), \quad (48)$$

which is composed of two parts: $E_1(\xi)$ and $E_2(\xi)$. The first part,

$$E_1(\xi) = \frac{1}{2}\xi^T \xi + \frac{1}{\beta} \ln(N) + \frac{1}{2}M^2 = \frac{1}{2}\xi^T \xi + C_1, \quad (49)$$

is a convex function, as it is a quadratic function in ξ plus a constant C_1 . The second part,

$$E_2(\xi) = -\frac{1}{\beta} \ln \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right), \quad (50)$$

is a concave function due to the negative logarithm of the sum of exponentials.

In the context of optimization, our goal is to minimize the sum $E_1 + E_2$. Given that E_1 is convex and E_2 is concave, we can apply the CCCP (Concave–Convex Procedure) algorithm. In this framework, $-E_2$ becomes a convex function, allowing us to define $u(x) = E_1(x)$ and $v(x) = -E_2(x)$ for the CCCP algorithm.

The update rule in the CCCP algorithm for this specific problem is then given by:

$$\xi^{(t+1)} \in \left(E_1(\xi) - \xi^T \nabla (-E_2(\xi^{(t)})) \right) = \left(E_1(\xi) + \xi^T \nabla (E_2(\xi^{(t)})) \right), \quad (51)$$

where $\nabla (E_2(\xi^{(t)}))$ represents the gradient of E_2 at $\xi^{(t)}$. This iterative process is repeated, and the resulting sequence satisfies a monotonic energy decrease as shown below.

We define an upper bound for the energy function, denoted as E_u . Let

$$E_u(\xi, \xi^{(t)}) = E_1(\xi) + E_2(\xi^{(t)}) + (\xi - \xi^{(t)})^T \nabla E_2(\xi^{(t)}). \quad (52)$$

This function is designed to be an upper bound for the original energy function. The rationale behind this definition is as follows:

- The term $E_1(\xi)$ is the first component of the original energy function, where the first argument ξ is used.
- The term $E_2(\xi^{(t)})$ is the second component of the energy function evaluated at $\xi^{(t)}$, the current iterate.
- The term $(\xi - \xi^{(t)})^T \nabla E_2(\xi^{(t)})$ represents the linear approximation of E_2 around $\xi^{(t)}$.

To prove that E_u is indeed an upper bound for $E(\xi)$, consider the case when $\xi = \xi^{(t)}$. In this scenario,

$$E_u(\xi^{(t)}, \xi^{(t)}) = E_1(\xi^{(t)}) + E_2(\xi^{(t)}) + (\xi^{(t)} - \xi^{(t)})^T \nabla E_2(\xi^{(t)}) = E_1(\xi^{(t)}) + E_2(\xi^{(t)}) = E(\xi^{(t)}). \quad (53)$$

Since $E_u(\xi, \xi^{(t)})$ equals $E(\xi)$ when $\xi = \xi^{(t)}$ and considering the properties of convex and concave functions (where E_1 is convex and E_2 is concave), it can be shown that $E_u(\xi, \xi^{(t)})$ serves as an upper bound for $E(\xi)$ for all ξ .

According to Boyd & Vandenberghe (2004) [22] in Section 3.2, a crucial condition for convexity of a differentiable function is presented. Suppose f is a differentiable function. Then, f is convex if and only if two conditions are satisfied:

- The domain of f , denoted as $\text{dom } f$, is a convex set.
- The inequality

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad (54)$$

holds for all x, y in $\text{dom } f$. The first condition ensures that the function f is defined over a convex set. This is a necessary condition because convexity involves comparing values of f at different points, and these comparisons only make sense if those points are within the domain where f is defined.

The second condition is known as the first-order condition for convexity. It states that at any point x in the domain of f , the function value at any other point y is greater than or equal to the value at x plus the inner product of the gradient of f at x and the vector $(y - x)$. This condition essentially means that the tangent (or linear approximation) at any point x lies below the graph of the function.

Let's consider the functions E_2 and E_u with respect to the convexity condition described by Boyd & Vandenberghe. By substituting $\xi = y$ and $\xi^{(t)} = x$ into the convexity condition, we analyze the behavior of $-E_2$. According to the convexity condition, for a convex function f , we have

$$f(y) \geq f(x) + \nabla f(x)^T (y - x). \quad (55)$$

Substituting $-E_2$ for f , this inequality becomes

$$-E_2(\xi) \geq -E_2(\xi^{(t)}) + \nabla(-E_2(\xi^{(t)}))^T (\xi - \xi^{(t)}). \quad (56)$$

Rearranging terms, we obtain

$$E_2(\xi) \leq E_2(\xi^{(t)}) - (\xi - \xi^{(t)})^T \nabla(-E_2(\xi^{(t)})). \quad (57)$$

This inequality shows that E_2 is concave, as $-E_2$ is convex.

Now, considering the energy function E and its upper bound E_u , we have

$$E(\xi) = E_1(\xi) + E_2(\xi). \quad (58)$$

Using the inequality derived for E_2 , we can state that

$$E(\xi) \leq E_1(\xi) + E_2(\xi^{(t)}) - (\xi - \xi^{(t)})^T \nabla(-E_2(\xi^{(t)})). \quad (59)$$

This leads to the conclusion that

$$E(\xi) \leq E_u(\xi, \xi^{(t)}) = E_1(\xi) + E_2(\xi^{(t)}) + (\xi - \xi^{(t)})^T \nabla E_2(\xi^{(t)}). \quad (60)$$

Therefore, $E_u(\xi, \xi^{(t)})$ serves as an upper bound for $E(\xi)$.

Considering the update rule for minimizing the energy function, we start with the observation that $\xi^{(t+1)}$ is chosen to minimize the upper bound $E_u(\xi, \xi^{(t)})$. Mathematically, this is expressed as

$$\xi^{(t+1)} \in E_u(\xi, \xi^{(t)}). \quad (61)$$

Therefore, we have

$$E_u(\xi^{(t+1)}, \xi^{(t)}) \leq E_u(\xi^{(t)}, \xi^{(t)}). \quad (62)$$

Given that $E_u(\xi, \xi^{(t)})$ is an upper bound for $E(\xi)$, it follows that

$$E(\xi^{(t+1)}) \leq E_u(\xi^{(t+1)}, \xi^{(t)}) \leq E_u(\xi^{(t)}, \xi^{(t)}) = E(\xi^{(t)}). \quad (63)$$

This inequality shows that the energy function E is non-increasing under this update rule. $\square$

Theorem 4 (Convergence of the CCCP iterates). *Consider the CCCP iteration associated with the energy function $E(\xi) = E_1(\xi) + E_2(\xi)$, and let $\{\xi^{(t)}\}_{t \geq 0}$ be generated by*

$$\xi^{(t+1)} = -X \text{Softmin}(\beta X^T \xi^{(t)}). \quad (64)$$

Then the sequence $\{\xi^{(t)}\}$ converges to a stationary point (a local minimum or a saddle point) of the energy function $E(\xi)$.

Proof of Theorem 4. By Theorem 3, the energy is monotonically non-increasing along the iterations:

$$E(\xi^{(t+1)}) \leq E(\xi^{(t)}), \quad \forall t \geq 0. \quad (65)$$

Moreover, from Theorem 1 (Non-negative) we have a lower bound

$$E(\xi^{(t)}) \geq 0, \quad \forall t \geq 0, \quad (66)$$

and from Theorem 2 (Upper boundedness) the energy is also bounded above. Hence the real-valued sequence $\{E(\xi^{(t)})\}$ is monotone and bounded, therefore it converges to some finite limit E^* .

Next, note that the update rule is derived by minimizing the CCCP upper bound (majorizer)

$$E_u(\xi, \xi^{(t)}) = E_1(\xi) + E_2(\xi^{(t)}) + (\xi - \xi^{(t)})^T \nabla E_2(\xi^{(t)}), \quad (67)$$

which satisfies

$$E(\xi) \leq E_u(\xi, \xi^{(t)}) \quad \text{for all } \xi, \quad (68)$$

and equality holds at $\xi = \xi^{(t)}$:

$$E(\xi^{(t)}) = E_u(\xi^{(t)}, \xi^{(t)}). \quad (69)$$

Thus, the CCCP update chooses $\xi^{(t+1)}$ as a minimizer of $E_u(\cdot, \xi^{(t)})$:

$$\xi^{(t+1)} \in \arg \min_{\xi} E_u(\xi, \xi^{(t)}). \quad (70)$$

Since E_1 is convex and $-E_2$ is convex, the CCCP framework applies and standard CCCP convergence theory [10,19–21] guarantees that every limit point of the sequence is a stationary point of $E(\xi)$; furthermore, under CCCP the iterates converge to a stationary point (local minimum or saddle point) of the original non-convex energy. Therefore, $\{\xi^{(t)}\}$ converges to a stationary point of $E(\xi)$. $\square$

3.5. Discussion

We want to establish a theoretically consistent energy formulation and optimization mechanism for the proposed softmin-based Hopfield network. Rather than introducing a new optimization framework, the Concave–Convex Procedure (CCCP) is employed as an analytical tool to formalize the update dynamics and to guarantee convergence properties of the resulting energy-based system.

By decomposing the proposed energy function $E(\xi)$ into convex and concave components, CCCP provides a principled way to construct a discrete-time dynamical system whose energy is guaranteed to decrease monotonically. This result ensures that the update rule derived in this work is not heuristic, but instead corresponds to a well-defined majorization–minimization procedure applied to a non-convex energy landscape. Importantly, CCCP is not introduced as a novel contribution, but rather serves to justify the optimization dynamics induced by the proposed energy function [10].

A key distinction of the present formulation lies in the use of the softmin operator within the energy function and the resulting update rule. Structurally, the proposed model remains closely related to modern Hopfield networks formulated with log-sum-exp energies. The primary modification is the sign change inside the exponential term, which leads to a softmin-based aggregation instead of the commonly used softmax. This change alters the geometry of the energy landscape and the direction of attraction during associative retrieval, while preserving the overall energy-based interpretation.

From an optimization perspective, the derived update rule

$$\xi^{t+1} = -X \text{Softmin}(\beta X^T \xi^t) \quad (71)$$

is shown to be the exact minimizer of the CCCP upper bound at each iteration. Consequently, the sequence of energy values is guaranteed to be non-increasing, and convergence to a stationary point of the energy function follows directly from standard CCCP theory. This result establishes that the proposed dynamics are stable and well-defined for arbitrary initial conditions.

It is important to emphasize that this section focuses exclusively on theoretical consistency and convergence properties. Issues such as classification accuracy, empirical robustness, or performance comparisons are deliberately deferred to later sections. The contribution of this chapter is to demonstrate that replacing the softmax operator with softmin within a modern Hopfield-style energy formulation yields a mathematically sound associative memory model with provable optimization guarantees.

We show that the proposed softmin-based Hopfield network admits a valid energy formulation and a convergent optimization procedure. The use of CCCP ensures that the update dynamics are theoretically grounded, while the softmin modification provides an alternative retrieval mechanism within the broader family of modern Hopfield networks.

Although our theoretical development is general, the proposed CCCP-grounded dynamics are directly relevant to practical Hopfield-based iterative inference modules. In modern architectures (e.g., Hopfield blocks used for refinement), the network state is updated multiple times to progressively improve predictions under partially observed, noisy, or ambiguous inputs. In such settings, a monotone energy descent and convergence guarantee provide a principled stability mechanism that is often missing when heuristic

updates are used. Moreover, the proposed softmin aggregation induces a conservative retrieval behavior that can be advantageous under overlap and heterogeneous pattern structures, aligning with the empirical gains observed in challenging simulated regimes and the stable performance on real benchmark datasets in Section 4.5.

Additional proofs of convexity and concavity are provided in Appendix A.

4. Simulation and Empirical Evaluation

We evaluate the proposed softmin-based Hopfield update rule through controlled simulation and classification experiments. The goal of these experiments is not to establish state-of-the-art performance, but to assess the empirical behavior of the proposed energy formulation and its convergence properties when embedded in a practical learning framework.

To isolate the effect of the proposed update dynamics, all experiments are conducted within a fixed architectural backbone. Performance differences observed in the following experiments can therefore be attributed to the modification of the Hopfield update rule rather than changes in model structure or training protocol.

4.1. Algorithm Structure

In this section, we introduce the Hopular architecture as the experimental framework used to evaluate the proposed softmin-based Hopfield update rule. Hopular provides a modular and well-established structure in which Hopfield-type associative memory modules are explicitly separated from the surrounding embedding and prediction layers, making it particularly suitable for controlled algorithmic comparisons [23].

Hopular is built upon the principles of modern Hopfield networks and extends them to operate jointly on input features and targets, referred to as attributes. Importantly, the architecture allows the underlying Hopfield modules to be modified without altering the overall data flow or training procedure. This property enables us to directly replace the original Hopfield update rule with the proposed softmin-based formulation while keeping all other components unchanged.

The Hopular architecture consists of three main components: an embedding layer, a stack of Hopular blocks, and a summarization layer [23]. Each component plays a distinct role in transforming the input data and refining attribute predictions.

- **Embedding Layer:** Each attribute of the input sample is first encoded in an e -dimensional embedding space. Categorical attributes are transformed using one-hot encoding, while continuous attributes are normalized to zero mean and unit variance. Separate learnable embeddings preserve attribute identity and position, and the concatenation of all attribute embeddings forms the initial prediction vector.
- **Hopular Blocks:** The initial prediction vector is iteratively refined through a sequence of Hopular blocks. Each block contains two Hopfield modules, denoted by H_s and H_f , which update attribute predictions in an alternating manner using residual connections. In our experiments, these Hopfield modules are instantiated using the proposed softmin-based update rule, while the surrounding architecture remains unchanged.
- **Summarization Layer:** The summarization layer maps the refined attribute representations to their final predictions. Feature predictions and target predictions are produced simultaneously, completing the forward pass of the model.

By adopting Hopular as a fixed architectural backbone, the experimental results presented in the following sections isolate the effect of the proposed energy formulation and update dynamics. This design ensures that any observed performance differences can be attributed to the softmin-based Hopfield mechanism rather than architectural modifications.

All experiments are conducted using a fixed architectural backbone to isolate the effect of the proposed Hopfield update rule. Varying architectures would introduce additional confounding factors, making it difficult to attribute performance differences specifically to the energy formulation and optimization dynamics. This controlled design allows a fair and interpretable comparison focused on the contribution of the proposed method.

As shown in Algorithm 1, the forward pass of Hopular starts with masking and embedding, followed by L Hopular blocks with two Hopfield modules, and ends with the summarization layer.

Algorithm 1 Forward pass of Hopular

Require: Hopfield modules H_s and H_f , embedding layer E , summarization layer S , number of features d , number of Hopular blocks L , and original input sample $x \in \mathbb{R}^d$

```

1:  $x \leftarrow \text{Mask}(x)$ 
2:  $\xi \leftarrow E(x)$ 
3: for  $i = 1$  to  $L$  do
4:    $\xi \leftarrow \xi + H_s(\xi)$ 
5:    $\Xi \leftarrow \text{Reshape}(\xi^T)$ 
6:    $\Xi \leftarrow \Xi + H_f(\Xi)$ 
7:    $\xi \leftarrow \text{Reshape}(\Xi)^T$ 
8: end for
9:  $\xi \leftarrow S(\xi)$ 
  
```

4.2. Simulation Design and Settings

The simulation study is designed to evaluate the empirical performance of the proposed algorithm under controlled and interpretable conditions. By generating synthetic datasets with known distributional properties, we are able to systematically assess the behavior of the method across varying levels of complexity before applying it to real-world data.

Synthetic classification tasks are adopted to systematically evaluate the behavior of the proposed Hopfield dynamics under controlled and interpretable data-generating mechanisms. By design, synthetic datasets allow us to independently vary specific sources of difficulty, such as class overlap, feature correlation, and multimodality, which cannot be isolated in real-world benchmarks.

We distinguish between low-dimensional and high-dimensional simulation regimes, as these settings pose fundamentally different challenges for classification. Low-dimensional simulations serve as a baseline, where class structures are relatively interpretable and many classical methods perform well. High-dimensional simulations, on the other hand, introduce issues such as noise accumulation, feature sparsity, and overfitting, providing a more stringent test of robustness. Since the proposed method is derived from an energy minimization framework with proven convergence properties, evaluating its performance under increasing dimensionality is essential.

Binary classification datasets are generated from multiple distributions, including normal, overlapping Gaussian, uniform, and mixture distributions, covering a range of separability and structural complexity. For each dataset, samples are split into training and testing sets using an 80/20 ratio. Cross-validation is applied within the training set to tune model-specific hyperparameters. Classification accuracy is used as the primary evaluation metric, and cross-entropy loss is additionally reported to assess the stability of probabilistic predictions.

This simulation design provides a concise yet comprehensive framework for comparing the proposed method with existing classifiers across diverse distributional and dimensional settings.

4.3. Simulation with Low Dimension Data

To systematically evaluate the behavior of different classification methods under varying data structures, we design four simulation settings based on distinct underlying distributions: normal, overlapping Gaussian, uniform, and mixture distributions. For each scenario, we generate a binary classification dataset consisting of 1000 samples, with 500 samples per class and two-dimensional feature vectors.

In the normal distribution setting, both classes are generated from multivariate Gaussian distributions with symmetric but opposite mean vectors. Specifically, one class is centered at

$$\mu_1 = (1, 1),$$

while the other class is centered at

$$\mu_2 = (-1, -1).$$

Both classes share an identical covariance matrix Σ with unit variances and strong positive correlations, where the off-diagonal elements are set to 0.8. This configuration represents a relatively well-structured scenario with correlated features and clear class separation.

The overlapping Gaussian distribution setting increases classification difficulty by bringing the class means closer together. In this case, the two classes are centered at

$$\mu_1 = (0, 0) \quad \text{and} \quad \mu_2 = (1, 1),$$

respectively, while maintaining correlated covariance structures with off-diagonal elements equal to 0.5. This setup introduces substantial overlap between the two distributions, resulting in ambiguous decision regions where no clear linear or nonlinear boundary exists.

For the uniform distribution setting, data points are generated uniformly within bounded hyper-rectangular regions. One class is sampled from the interval

$$[0, 1]^2,$$

while the other class is sampled from

$$[0.5, 1.5]^2.$$

Unlike the Gaussian settings, this scenario lacks intrinsic clustering or density concentration, serving as a structure-free baseline for evaluating classifier robustness.

Finally, the mixture distribution setting introduces heterogeneity by sampling each class from a mixture of two Gaussian components with different mean vectors and covariance structures. This results in multimodal class distributions with varying correlations, reflecting complex data patterns in which class labels are not associated with a single dominant generating mechanism.

Figure 1 presents scatter plots for all four simulated scenarios. These visualizations provide an intuitive understanding of the geometric and statistical characteristics associated with each underlying distribution.

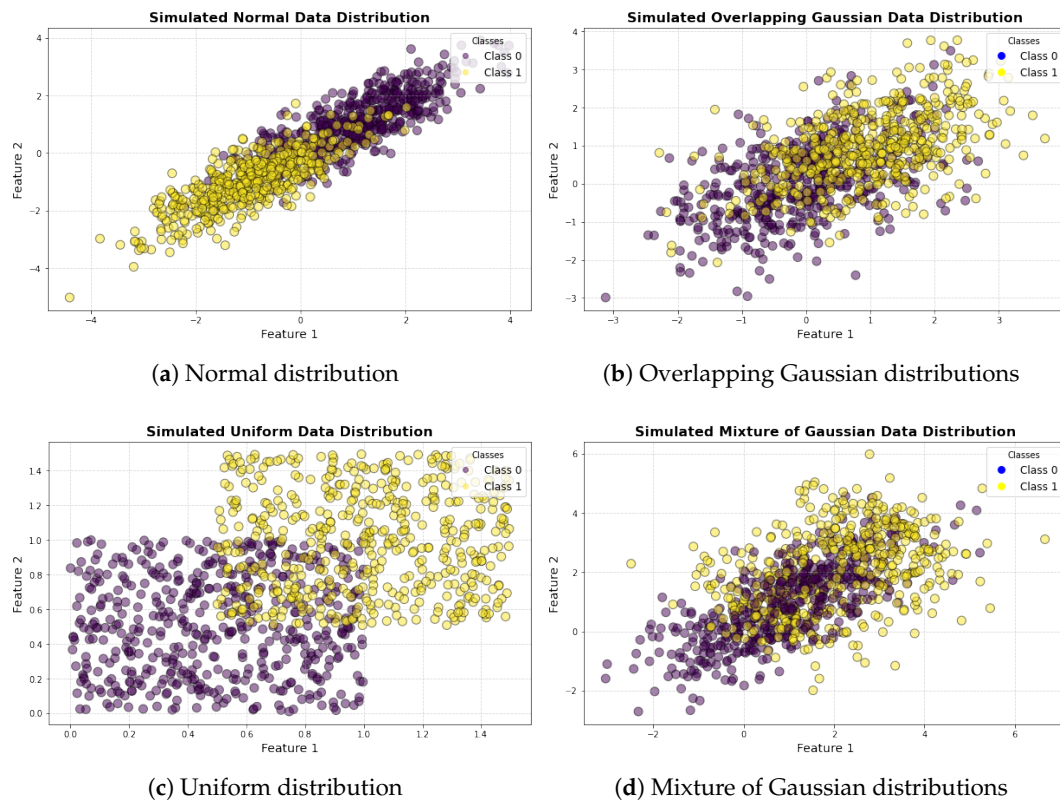


Figure 1. Visualization of simulated datasets under different distributional settings. **(a)** Normal distribution with well-separated Gaussian clusters. **(b)** Overlapping Gaussian distributions with reduced separation between class means. **(c)** Uniform distribution over bounded regions without intrinsic clustering structure. **(d)** Mixture of Gaussian distributions exhibiting multimodal and heterogeneous patterns.

The normal distribution scenario exhibits two well-separated elliptical clusters, reflecting clear class separation under correlated Gaussian features. In contrast, the overlapping Gaussian setting shows substantial inter-class mixing, resulting in ambiguous regions where class boundaries are difficult to distinguish. The uniform distribution displays evenly dispersed data points with minimal structural cues, indicating the absence of intrinsic clustering or density concentration. Finally, the mixture distribution reveals multiple sub-clusters within each class, illustrating multimodal and heterogeneous data structures.

Together, these scatter plots demonstrate a progression from structured and well-separated data to increasingly unstructured and heterogeneous settings. This progression motivates a comprehensive comparison of classification methods, as it allows for a systematic evaluation of model performance under varying degrees of complexity and distributional uncertainty.

Table 1 reports the classification accuracy of six methods across the four distributional settings. Several patterns emerge from this comparison. Across all distributions, the proposed Hopfield neural network achieves the highest classification accuracy. The performance gain is relatively modest for the normal distribution, where most classifiers perform reasonably well, but becomes increasingly pronounced as the data structure becomes more challenging. In particular, under overlapping Gaussian distributions, the Hopfield network demonstrates a substantial advantage, exceeding competing methods by a wide margin.

Table 1. Comparison of classification methods across different data distributions.

Distribution	SVM	Decision Tree	Random Forest	CNN	Ensemble	Hopfield
Normal	85.5%	78.5%	84.5%	80.0%	85.0%	86.5%
Uniform	86.0%	85.5%	89.0%	85.0%	89.0%	91.9%
Overlapping Gaussian	64.0%	60.0%	64.0%	65.0%	65.0%	73.5%
Mixture	67.5%	62.5%	69.0%	65.0%	66.0%	72.1%

To complement the point estimates in Table 1, we report 95% confidence intervals (CIs) for accuracy under each distributional setting. Since each simulated dataset contains $n = 1000$ labeled samples, we treat the observed accuracy $\hat{p}$ as a binomial proportion and compute the Wilson 95% CI. Table 2 summarizes the proposed Hopfield network versus the strongest baseline in each setting. Notably, under the most challenging overlapping Gaussian setting, the 95% CIs do not overlap, indicating the gain is unlikely to be explained by finite-sample variability alone.

Table 2. Accuracy with 95% Wilson confidence intervals (Hopfield vs. best baseline) for low-dimensional simulations ($n = 1000$).

Distribution	Best Baseline	Baseline Acc. (95% CI)	Hopfield Acc. (95% CI)	Δ Acc.
Normal	SVM	85.5% [83.2, 87.5]	86.5% [84.2, 88.5]	+1.0%
Uniform	Random Forest	89.0% [86.9, 90.8]	91.9% [90.0, 93.4]	+2.9%
Overlapping Gaussian	CNN	65.0% [62.0, 67.9]	73.5% [70.7, 76.1]	+8.5%
Mixture	Random Forest	69.0% [66.1, 71.8]	72.1% [69.2, 74.8]	+3.1%

These results suggest that while traditional classifiers perform adequately under favorable conditions, their performance degrades rapidly in the presence of overlap, lack of structure, or multimodality. In contrast, the Hopfield-based approach exhibits greater robustness across diverse data-generating mechanisms.

Our four low-dimensional settings are designed to isolate specific sources of heterogeneity. The Gaussian settings introduce strong feature correlations (off-diagonal covariance 0.8 and 0.5), while the mixture setting introduces multimodality by combining multiple Gaussian components per class. As shown in Table 2, the performance gap between Hopfield and the baselines increases as the data become more heterogeneous (overlap and multimodality), suggesting that the proposed energy-based retrieval mechanism is less sensitive to these complexities than standard classifiers.

Importantly, the Hopfield model remains competitive across four qualitatively different data-generating mechanisms (correlated Gaussian, overlapping Gaussian, structure-free uniform, and multimodal mixtures), suggesting the improvement is not confined to a single favorable setting.

In addition to accuracy and confidence interval, we evaluate model performance using cross entropy loss, which provides a probabilistic measure of prediction quality. Unlike accuracy, cross entropy captures not only whether predictions are correct, but also how confident the model is in its predictions.

For a binary classification dataset

$$\{(x_i, y_i)\}_{i=1}^N,$$

where $y_i \in \{0, 1\}$ denotes the true label of the i -th observation and $\hat{y}_i$ represents the predicted probability of class 1, the cross entropy loss is defined as

$$L = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)].$$

For multi-class classification with K classes, where y_{ik} is the one-hot encoded true label indicating whether sample i belongs to class k , and $\hat{y}_{ik}$ denotes the predicted probability of class k , the cross entropy loss generalizes to

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log(\hat{y}_{ik}).$$

Cross entropy loss is particularly suitable for comparing classifiers in simulation studies, as it penalizes overconfident incorrect predictions and provides a smooth objective for optimization and evaluation.

Table 3 presents the cross entropy test loss for all methods under each distribution. The Hopfield neural network consistently achieves the lowest or near-lowest test loss across all scenarios.

Table 3. Comparison of test loss across different data distributions.

Distribution	SVM	Decision Tree	Random Forest	CNN	Ensemble	Hopfield
Normal	0.33	2.20	0.5037	0.45	0.38	0.331
Uniform	0.33	1.80	0.27	0.29	0.26	0.25
Overlapping Gaussian	0.58	3.10	0.9894	0.80	0.75	0.56
Mixture	0.61	1.20	0.6275	0.68	0.60	0.54

Notably, in the overlapping Gaussian and mixture distribution settings, where uncertainty and heterogeneity are highest, the Hopfield network maintains a clear advantage in terms of loss reduction. This indicates that its probabilistic outputs are better calibrated and more stable compared to traditional classifiers, which often exhibit higher loss due to overfitting or poorly calibrated decision boundaries.

Overall, the consistency between accuracy improvements and reduced test loss reinforces the conclusion that the proposed Hopfield neural network with the softmax update rule provides both improved classification performance and more reliable probabilistic predictions across a wide range of data distributions.

4.4. Simulation with High Dimension Data

We investigated the performance of the Hopfield network when applied to high-dimensional data. Compared to the low-dimensional settings discussed previously, we increase the number of variables from $p = 2$ to $p = 50$, and simultaneously enlarge the sample size from $N = 1000$ to $N = 2000$. This increase in dimensionality introduces a substantially more challenging learning environment, where the advantages of Hopfield-type associative mechanisms may become less pronounced.

The reduced effectiveness in high-dimensional spaces is closely related to the intrinsic storage and capacity limitations of Hopfield networks. As dimensionality increases, the concentration of measure phenomenon and increased pattern interference can negatively impact the stability of attractor states, making reliable retrieval more difficult.

To mitigate the adverse effects of high dimensionality and to facilitate visualization, we apply Principal Component Analysis (PCA) as a preprocessing step prior to classification. PCA is a widely used dimensionality reduction technique that projects the original data

into a lower-dimensional subspace while preserving as much variance as possible. In this study, PCA is employed to project the high-dimensional data onto a two-dimensional space, primarily for visualization and exploratory analysis rather than performance enhancement.

Figure 2 illustrates the PCA-projected scatter plots for the high-dimensional normal and uniform distribution settings. Despite the dimensionality reduction, differences in the underlying data structures remain observable. The normal distribution retains a relatively coherent cluster structure, whereas the uniform distribution exhibits a more diffuse and structure-free pattern, reflecting the inherent difficulty of the classification task.

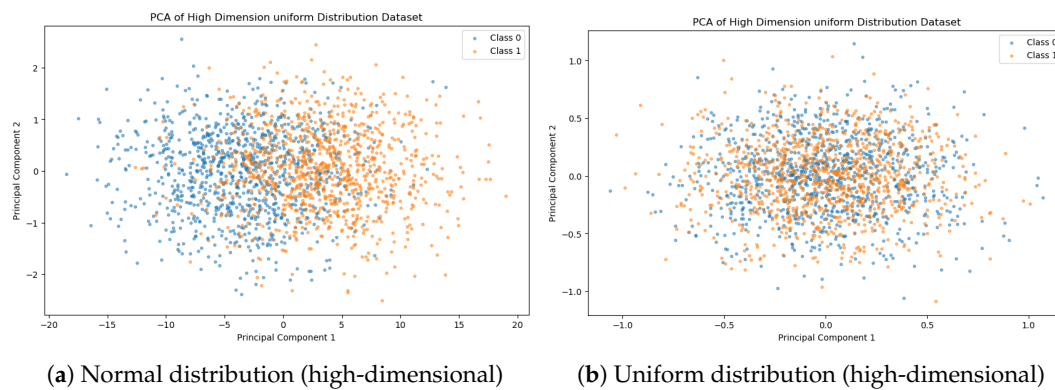


Figure 2. PCA-based visualization of high-dimensional data under different distributional settings. (a) High-dimensional normal distribution projected onto the first two principal components. (b) High-dimensional uniform distribution projected onto the first two principal components.

Table 4 reports the classification accuracy of all considered methods on the high-dimensional datasets. Compared to the low-dimensional scenarios, overall accuracy decreases across most classifiers, highlighting the increased difficulty induced by higher dimensionality. We apply PCA only for visualization of the high-dimensional data; all classification results in Table 4 are obtained using the original 50-dimensional features for every method. Table 4 reports results in the original 50D space (PCA is used only for visualization).

Table 4. Comparison of classification methods on high-dimensional data.

Distribution	SVM	Decision Tree	Random Forest	CNN	Ensemble	Hopfield
Normal (High-Dim)	75.5%	65.3%	73.75%	72.1%	74.3%	73.5%
Uniform (High-Dim)	60.5%	60.2%	76.0%	75.2%	76.1%	75.0%

For the high-dimensional normal distribution, SVM achieves the highest classification accuracy, followed closely by ensemble and random forest methods. Although the Hopfield network does not outperform all competing approaches in this setting, its performance remains competitive, indicating that the network can still capture meaningful structure after dimensionality reduction.

In the high-dimensional uniform distribution scenario, ensemble and random forest methods exhibit the strongest performance, reflecting their robustness in structure-free environments. The Hopfield network achieves accuracy comparable to that of CNN, demonstrating resilience despite the absence of strong geometric structure. In contrast, SVM and decision tree classifiers experience a notable decline in performance under this distribution.

Overall, these results suggest that while the Hopfield network's relative advantage diminishes as dimensionality increases, it continues to provide stable and competitive performance. When combined with appropriate dimensionality reduction techniques such

as PCA, the Hopfield network remains a viable classification approach for high-dimensional data, particularly in settings where interpretability and associative memory properties are desirable.

Additional implementation details are provided in Appendix B.

4.5. Real-World Benchmark Evaluation

To evaluate the proposed method on real-world data, we consider several widely used benchmark datasets spanning medical diagnosis, text classification, and public health analytics. These datasets have been extensively studied in the literature, and well-established experimental settings exist for commonly used classifiers such as SVM, random forest, CNN, and ensemble methods.

- **Diabetes Dataset:** A clinical benchmark dataset for binary diabetes prediction based on physiological measurements, frequently used to evaluate statistical and machine learning classifiers [24–26].
- **Sports Article Dataset:** A text classification dataset consisting of sports articles labeled as subjective or objective, commonly adopted in natural language processing research [27,28].
- **COVID-19 Dataset:** A large-scale county-level dataset integrating demographic, geographic, and health-related information for modeling COVID-19 cases and mortality in the United States [29–31].

These datasets are chosen because they are well-established benchmarks with extensive prior use in the literature. For each dataset, strong baseline performance has already been demonstrated using traditional and modern classifiers, including SVM, random forest, CNN, and ensemble methods. As a result, they provide a reliable and fair testbed for assessing the effectiveness and generalizability of the proposed Hopfield-based approach without requiring extensive task-specific tuning.

Table 5 summarizes the classification accuracy of different methods on real-world datasets drawn from multiple application domains. Overall, the proposed Hopfield neural network demonstrates competitive and stable performance across all datasets, particularly in settings characterized by complex feature interactions.

Table 5. Classification accuracy (%) on real-world datasets.

Method	Sports	Diabetes	COVID-19 (Death Ratio)	COVID-19 (Case Ratio)
SVM	79.50	78.45	74.00	68.00
Random Forest	78.80	76.20	73.20	64.50
Decision Tree	79.90	75.50	73.80	66.90
Conventional Neural Network	82.87	79.80	74.90	67.50
Parallel Ensemble	82.00	75.00	76.00	68.30
Sequential Ensemble	85.34	79.40	77.20	69.60
Hopfield Neural Network	84.96	77.00	77.60	71.00

On the sports article dataset, the Hopfield network achieves an accuracy of 84.96%, which is slightly lower than that of the sequential ensemble method but higher than all remaining classifiers. This result indicates that the Hopfield network is effective in capturing nontrivial patterns in high-dimensional textual features without relying on extensive ensemble structures.

For the diabetes dataset, the Hopfield network attains an accuracy of 77.00%, remaining competitive with conventional neural networks and ensemble-based approaches. Although it does not achieve the highest accuracy in this setting, its performance suggests robustness in modeling nonlinear relationships among clinical features.

On the COVID-19 datasets, the Hopfield network achieves the best performance in both death ratio prediction (77.60%) and case ratio prediction (71.00%). These results highlight its advantage in handling large-scale, heterogeneous data involving demographic, geographic, and health-related variables. The energy-based formulation of the Hopfield network enables stable convergence in the presence of correlated and noisy real-world features.

As shown in Table 6, we report 95% Wilson confidence intervals to quantify the uncertainty of test accuracy from a finite test set. For the COVID-19 (Case Ratio) dataset, Hopfield improves over the best baseline by 1.40%, while for the COVID-19 (Death Ratio) dataset the improvement is 0.40%; however, the confidence intervals overlap, suggesting that these real-data gains should be interpreted as modest rather than decisively significant under a single split. This motivates our use of mechanism-controlled simulations, where distributional factors can be varied systematically to better understand robustness trends.

Table 6. Accuracy with 95% Wilson confidence intervals (Hopfield vs. best baseline) for real-world datasets under a single stratified 80/20 split.

Dataset (Test Size)	Best Baseline	Baseline Acc. (95% CI)	Hopfield Acc. (95% CI)	Δ Acc.
Sports	Sequential Ensemble	85.34% [79.77, 89.58]	84.96% [79.35, 89.25]	−0.38%
Diabetes	CNN	79.80% [72.77, 85.38]	77.00% [69.74, 82.94]	−2.80%
COVID(Death Ratio)	Sequential Ensemble	77.20% [73.76, 80.31]	77.60% [74.18, 80.69]	+0.40%
COVID(Case Ratio)	Sequential Ensemble	69.60% [65.89, 73.07]	71.00% [67.33, 74.41]	+1.40%

Robustness is reflected by performance stability across heterogeneous data structures. As shown in Table 1, the Hopfield model maintains a high accuracy not only in the well-separated Gaussian case but also under substantial overlap and multimodality, where several baselines degrade more noticeably. Together with the real-world results in Table 5, this suggests the proposed approach remains practically useful under non-ideal, heterogeneous conditions.

Taken together, these results demonstrate that the proposed Hopfield neural network provides a strong and reliable alternative to conventional classifiers on real-world datasets, particularly in complex and high-dimensional settings where feature interactions and uncertainty play a significant role.

5. Discussion and Conclusions

This work presents a comprehensive empirical evaluation of a modified Hopfield neural network equipped with a softmin-based update rule. Through systematic simulations under diverse distributional settings and dimensional regimes, as well as experiments on multiple real-world datasets, we demonstrate that the proposed formulation yields improved classification performance and more stable probabilistic predictions compared to conventional classifiers.

The softmin update rule plays a central role in enhancing robustness, particularly in scenarios involving overlapping classes, heterogeneous structures, and noisy observations. By reshaping the energy landscape and emphasizing the minimization of large similarity values, the proposed Hopfield network exhibits superior stability and improved calibration, as reflected by consistent reductions in cross entropy loss. These advantages become especially pronounced in low-dimensional and moderately complex settings, where traditional methods often struggle with uncertainty and multimodality.

At the same time, our results indicate that the relative advantage of the Hopfield network diminishes as dimensionality increases. This behavior is closely related to intrinsic capacity and interference limitations of associative memory models. Nevertheless, even in

high-dimensional scenarios, the proposed method remains competitive and exhibits stable performance when combined with appropriate dimensionality reduction techniques.

Overall, this study highlights the effectiveness of integrating energy-based optimization with a softmin update rule in modern Hopfield networks. The proposed approach provides a principled and robust alternative for classification tasks involving complex feature interactions and uncertainty. Future work will focus on improving scalability in high-dimensional settings and exploring hybrid architectures that further extend the applicability of Hopfield-type models.

Author Contributions: Conceptualization, Y.B.; Methodology, Y.B.; Software, Y.B.; Formal analysis, Y.B.; Validation, Y.B.; Visualization, Y.B.; Writing—original draft preparation, Y.B.; Writing—review and editing, Y.B., M.A., T.D.L., T.J., J.H., Y.X. and D.H.; Investigation, Y.B., M.A. and T.D.L.; Resources, T.J., J.H., Y.X. and D.H.; Data curation, Y.B., T.J. and J.H.; Project administration, Y.B. and M.A.; Supervision, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets analyzed in this study are publicly available and have been cited in the manuscript.

Acknowledgments: The authors primarily prepared this manuscript through manual writing. Generative artificial intelligence tools were used in a limited manner to assist with language polishing and formatting. The authors reviewed and edited all content and take full responsibility for the final version of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Convexity and Concavity Verification

To verify the convexity and concavity of $E_1(\xi)$ and $E_2(\xi)$, we use standard mathematical tools such as the second-order derivative test and the Hessian matrix. By analyzing the curvature of these functions, we can determine whether they are convex or concave.

Theorem A1. *The function*

$$E_1(\xi) = \frac{1}{2}\xi^T \xi + C_1$$

is convex with respect to ξ .

Proof of Theorem A1. To verify the convexity of $E_1(\xi)$, we show that it satisfies the definition of convexity:

$$E(\theta\xi_1 + (1-\theta)\xi_2) \leq \theta E(\xi_1) + (1-\theta)E(\xi_2), \quad (\text{A1})$$

for all $\xi_1, \xi_2 \in \mathbb{R}$ and $\theta \in [0, 1]$.

Substituting the expressions of $E(\xi_1)$ and $E(\xi_2)$, we obtain

$$\frac{1}{2}(\theta\xi_1 + (1-\theta)\xi_2)^2 + C_1 \leq \theta\left(\frac{1}{2}\xi_1^2 + C_1\right) + (1-\theta)\left(\frac{1}{2}\xi_2^2 + C_1\right). \quad (\text{A2})$$

Expanding the quadratic terms and simplifying yields

$$\frac{1}{2}(\theta^2\xi_1^2 + 2\theta(1-\theta)\xi_1\xi_2 + (1-\theta)^2\xi_2^2) + C_1 \leq \frac{1}{2}\theta\xi_1^2 + \frac{1}{2}(1-\theta)\xi_2^2 + C_1. \quad (\text{A3})$$

Canceling common terms and rearranging, we obtain

$$(\theta \xi_1 - (1 - \theta) \xi_2)^2 \geq 0. \quad (\text{A4})$$

Since this inequality always holds, $E_1(\xi)$ is convex. $\square$

Theorem A2. *The function*

$$E_2(\xi) = -\frac{1}{\beta} \ln \left(\sum_{i=1}^N \exp(-\beta x_i^T \xi) \right)$$

is concave with respect to ξ .

Proof of Theorem A2. To prove the concavity of $E_2(\xi)$, it suffices to show that its Hessian matrix is negative semidefinite. We first recall the following classical result.

Theorem A3. *The log-sum-exp function f , defined by*

$$f(\mathbf{x}) = \log \left(\sum_{i=1}^n e^{x_i} \right),$$

is convex.

Proof. The monotonicity of the log-sum-exp function follows from the monotonicity of the exponential and logarithmic functions. To prove convexity, we examine the Hessian of f , given by

$$\nabla^2 f(\mathbf{x}) = \frac{1}{S(\mathbf{x})^2} \left(S(\mathbf{x}) \text{diag}(\mathbf{s}(\mathbf{x})) - \mathbf{s}(\mathbf{x}) \mathbf{s}(\mathbf{x})^T \right),$$

where $\mathbf{s}(\mathbf{x}) = (e^{x_1}, \dots, e^{x_n})$ and $S(\mathbf{x}) = \sum_{i=1}^n s_i(\mathbf{x})$.

For any vector $\mathbf{z} \in \mathbb{R}^n$, we have

$$\begin{aligned} S(\mathbf{x})^2 \mathbf{z}^T \nabla^2 f(\mathbf{x}) \mathbf{z} &= \left(\sum_{i=1}^n s_i(\mathbf{x}) z_i^2 \right) \left(\sum_{i=1}^n s_i(\mathbf{x}) \right) - \left(\sum_{i=1}^n s_i(\mathbf{x}) z_i \right)^2 \\ &\geq 0, \end{aligned}$$

where the inequality follows from the Cauchy–Schwarz inequality. Thus, f is convex. $\square$

Since $E_2(\xi)$ differs from the log-sum-exp function by a negative sign and a positive scaling factor $1/\beta$, it follows that $-E_2(\xi)$ is convex. Consequently, the Hessian of $E_2(\xi)$ is negative semidefinite, and $E_2(\xi)$ is concave with respect to ξ . $\square$

Appendix B. Comparison of Softmax and Softmin in Outlier-Contamination Simulations

We present a concise comparison between the softmax (original modern Hopfield/Hopular-style) and the proposed softmin update, and outline an outlier-contamination simulation used to evaluate their robustness.

For each run, we generate a clean binary dataset with $n = 1000$ and $d = 2$ from a Gaussian model (including the overlapping-Gaussian and Gaussian-mixture settings in Figure 1). Given an outlier proportion $\pi \in \{1.6\%, 3\%, 5\%, 10\%, 20\%\}$, we randomly select

$\lfloor \pi n \rfloor$ samples and replace their feature vectors by Gaussian outliers drawn from the same mean structure but with inflated covariance:

$$x_i \leftarrow x_i^{\text{out}}, \quad x_i^{\text{out}} \sim \mathcal{N}(\mu_i, \Sigma_i),$$

where (μ_i, Σ_i) follows the generating component of the corresponding clean Gaussian setting. This produces a controlled fraction of extreme points while keeping the overall geometry comparable across π (Figure A1).

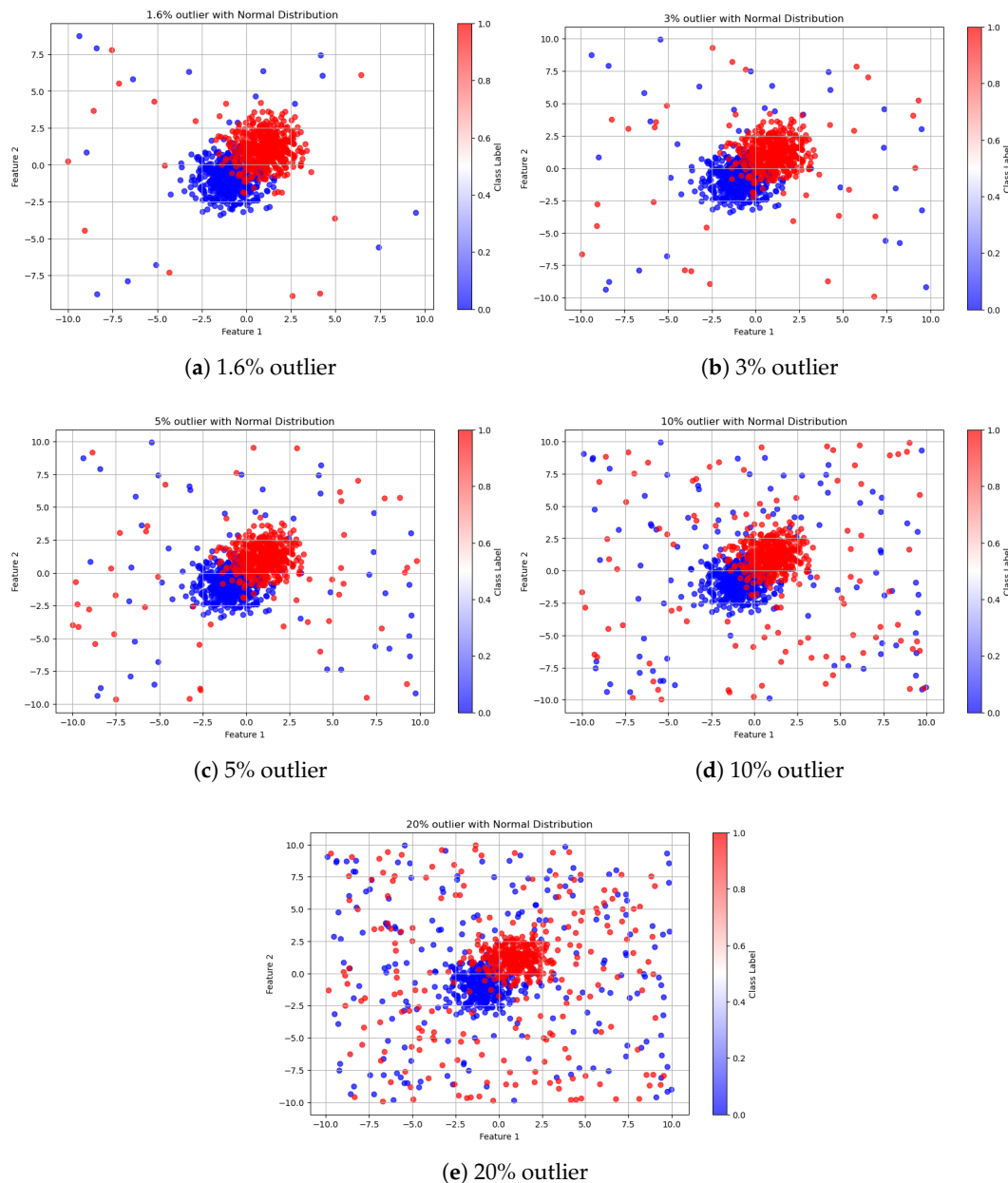


Figure A1. Synthetic binary dataset with increasing outlier proportions. As the outlier proportion increases, the distribution becomes increasingly dominated by extreme points, visually obscuring the original separation structure.

Both updates compute weights from scores $z_i = x_i^\top \zeta$ and aggregate patterns through X . Softmax assigns weights proportional to $\exp(\beta z_i)$, so a few outliers that induce unusually large z_i can dominate the update. Softmin uses weights proportional to $\exp(-\beta z_i)$, which suppresses such large scores and reduces the influence of extreme samples. Therefore,

under increasing contamination, softmax tends to yield a more stable retrieval step and slower performance degradation.

Figure A2 reports the classification accuracy as the outlier proportion increases from 1.6% to 20%. The updated (softmax-based) method remains consistently higher and degrades more slowly than the original (softmax-based) baseline, indicating improved robustness under heavier outlier contamination.

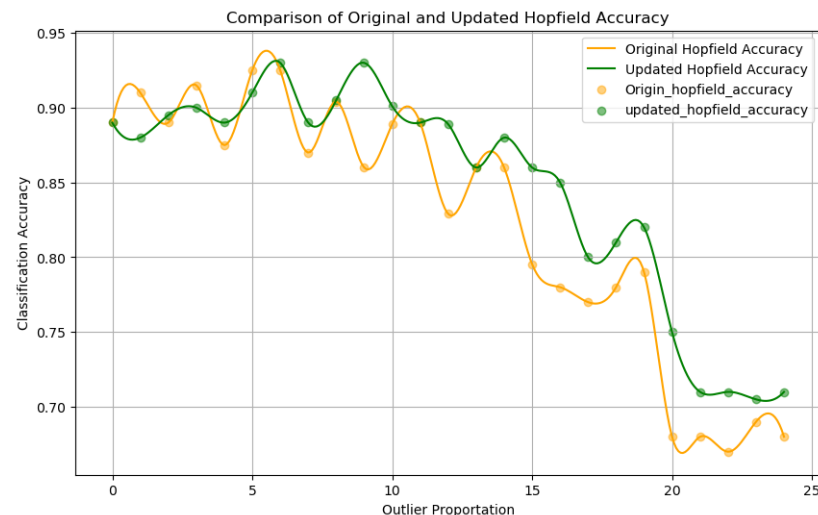


Figure A2. Classification accuracy versus outlier proportion for the original (softmax-based) and updated (softmax-based) Hopfular algorithms.

References

1. Ramsauer, H.; Schäfl, B.; Lehner, J.; Seidl, P.; Widrich, M.; Adler, T.; Gruber, L.; Holzleitner, M.; Pavlović, M.; Sandve, G.K.; Greiff, V. Hopfield networks is all you need. *arXiv* **2020**, arXiv:2008.02217.
2. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558.
3. Hopfield, J.J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA* **1984**, *81*, 3088–3092.
4. Amit, D.J.; Gutfreund, H.; Sompolinsky, H. Statistical mechanics of neural networks near saturation. *Ann. Phys.* **1987**, *173*, 30–67.
5. Amit, D.J.; Gutfreund, H.; Sompolinsky, H. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.* **1985**, *55*, 1530–1533.
6. Krotov, D.; Hopfield, J.J. Dense associative memory for pattern recognition. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1172–1180.
7. Hertz, J.; Krogh, A.; Palmer, R.G.; Horner, H. *Introduction to the Theory of Neural Computation*; Addison-Wesley: Redwood City, CA, USA, 1991.
8. Demircigil, M.; Heusel, J.; Löwe, M.; Upgang, S.; Vermet, F. On a model of associative memory with huge storage capacity. *J. Stat. Phys.* **2017**, *168*, 288–299.
9. Bao, Y. Hopfield Neural Networks: An Updated Approach for Using Associative Memory to Improve Matrices. Ph.D. Dissertation, Auburn University, Auburn, AL, USA, 2024.
10. Yuille, A.L.; Rangarajan, A. The concave–convex procedure. *Neural Comput.* **2003**, *15*, 915–936.
11. Amit, D.J.; Gutfreund, H.; Sompolinsky, H. Information storage in neural networks with low levels of activity. *Phys. Rev. A* **1987**, *35*, 2293–2303.
12. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
13. LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; Huang, F. A tutorial on energy-based learning. In *Predicting Structured Data*; Bakir, G., Hofmann, T., Schölkopf, B., Smola, A., Eds.; MIT Press: Cambridge, MA, USA, 2006.
14. Ackley, D.H.; Hinton, G.E.; Sejnowski, T.J. A learning algorithm for Boltzmann machines. *Cogn. Sci.* **1985**, *9*, 147–169.
15. Hinton, G.E.; Sejnowski, T.J. Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*; Rumelhart, D.E., McClelland, J.L., Eds.; MIT Press: Cambridge, MA, USA, 1986; Volume 1, pp. 282–317.

16. Choromanska, A.; Henaff, M.; Mathieu, M.; Arous, G.B.; LeCun, Y. The loss surfaces of multilayer networks. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, PMLR, San Diego, CA, USA, 9–12 May 2015; pp. 192–204.
17. Jordan, M.I.; Ghahramani, Z.; Jaakkola, T.S.; Saul, L.K. An introduction to variational methods for graphical models. *Mach. Learn.* **1999**, *37*, 183–233.
18. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
19. Rangarajan, A.; Gold, S.; Mjolsness, E. A novel optimizing network architecture with applications. *Neural Comput.* **1996**, *8*, 1041–1060.
20. Rangarajan, A.; Yuille, A.L.; Mjolsness, E. Convergence properties of the softassign quadratic assignment algorithm. *Neural Comput.* **1999**, *11*, 1455–1474.
21. Lanckriet, G.; Sripierumbudur, B.K. On the convergence of the concave–convex procedure. *Adv. Neural Inf. Process. Syst.* **2009**, *22*, 1759–1767.
22. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
23. Schäfl, B.; Gruber, L.; Bitto-Nemling, A.; Hochreiter, S. Hoplar: Modern Hopfield networks for tabular data. *arXiv* **2022**, arXiv:2206.00664.
24. Bao, Y.; Wang, Z.R.; Guo, J.; Gaillard, P. Special Plots Methods with Diabetes Disease Data. In *Proceedings of PharmaSUG 2020; Data Visualization and Reporting (DV-163)*; San Francisco, CA, USA, 10–13 May 2020; pp. 1–10.
25. UCIML. Pima Indians Diabetes Database. *Kaggle Datasets* 2017. Available online: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database> (accessed on 25 February 2026).
26. Islam, M.; Ferdousi, R.; Rahman, S.; Bushra, H.Y. Likelihood prediction of diabetes at early stage using data mining techniques. In *Computer Vision and Machine Intelligence in Medical Image Analysis*; Springer: Singapore, 2020; pp. 113–125. https://doi.org/10.1007/978-981-13-8798-2_12.
27. Bao, Y.; Gaillard, P. Summarizing some conventional methods to classify a binary target. In *Proceedings of the SESUG 2019 Conference*; LexJansen, Inc.: Williamsburg, VA, USA, 2019.
28. Rizk, Y.; Awad, M. Sports articles for objectivity analysis. *Uci Mach. Learn. Repos.* **2018**. <https://doi.org/10.24432/C5801R>.
29. Bao, Y.; Gaillard, P. Potential Variables Analysis Affects Covid-19 Spread. In *Proceedings of PharmaSUG 2022*, Austin, TX, USA, 22–25 May 2022; pp. 1–11. Artificial Intelligence (Machine Learning) (AI-116). Available online: <https://ssrn.com/abstract=5806882> (accessed on 25 February 2026).
30. Bao, Y.; Huang, I.; Li, Q.; Zhang, Z.; Xing, Y.; Hou, D.; Ye, J. A framework for modeling county-level COVID-19 transmission. *Front. Public Health* **2025**, *13*, 1608360.
31. Haratian, A.; Fazelinia, H.; Maleki, Z.; Ramazi, P.; Wang, H.; Lewis, M.A.; Greiner, R.; Wishart, D. Dataset of COVID-19 outbreak and potential predictive features in the USA. *Data Brief* **2021**, *38*, 107360.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.