Prospectus DRAFT

Connor Hanley

October 8, 2025

Abstract

Dense Associative Memories generalize traditional Hopfield networks, and provide a mechanism to drastically improve their theoretical storage capacity. To do this, they perform a non-linear operation on the similarities between cue or query patterns with memorized patterns. These improvements come at the cost of the simple intuitions of Hopfield networks. We discuss contemporary and historical research in the field, and provide desiderata for a Hebbian model which retains the intuition and simplicity of Hopfield networks, without sacrificing theoretical storage capacity.

1 Introduction

Humans are able to recognize and retrieve patterns of data using distorted, noisy, and partial patterns (Rumelhart et al., 1986). This capacity of human memory is known as *content-addressability*: patterns which are stored in memory are able to be “looked up” by themselves or their parts. Modeling this property is a classical task in computational cognitive and neuroscience (see Amari (1972), Little (1974), Marr (1971), Nakano (1972), and Stanley (1976)). The family of models which implement content-addressability are known as *associative memory models* (AMs). AMs are able to account for many cognitive tasks, (INSERT EXAMPLES HERE) (Hintzman, 1984). A recent revival of interest AMs in machine learning research, driven by their equivalence with “attention” layers in the transformer architecture (Ramsauer et al., 2021; Vaswani et al., 2023), has led to drastic advances in the storage capacity of AMs (Demircigil et al., 2017; Hu et al., 2024; Krotov & Hopfield, 2016).

Traditional models of Associative Memory rely on Hebbian update rules to memorize stored patterns in the weights of lateral connections (Amari, 1972; Hopfield, 1982; Kohonen, 1988; Nakano, 1972). Hebbian update rules are appealing for cognitive and neuroscience as the weight updates are both entirely *local* and the representation of content is *distributed.* By local, we mean that the interactions between artificial neurons are entirely defined by local connections between other neurons. By distributed content, we mean that the representations of features, stimuli, and patterns are not defined in terms of single memory locations (like conventional digital computers), but rather are represented in the interactions over many artificial neurons (for motivation see McClelland et al. (1986)).

While traditional models have desirable properties, they suffer from small storage capacities (Amit et al., 1985; Hopfield, 1982), scaling linearly with the dimension of patterns stored. Dense Associative Memories have generalized traditional models, providing a theoretical storage capacity which scales exponentially with the dimension of the patterns stored (Demircigil et al., 2017; Krotov & Hopfield, 2016). This discovery has been incredibly fruitful, leading to increased interpretability of large language models based on the Transformer architecture (Ramsauer et al., 2021; Vaswani et al., 2023), as well as the shift from memorization to generalization in modern deep learning architectures (Pham et al., 2025). However, in spite of these gains, the Hebbian intuition and plausibility is lost (McAlister et al., 2025). Instead of relying on local update rules, Dense Associative Memories train using back-propagation of error.

In the following, we will investigate what a Hebbian alternative to Dense Associative Memory must look like in order to maintain biological plausibility and inherent interpretability without sacrificing theoretical capacity. To begin, we will review traditional Associative Memory models, starting with models similar to or based on Hopfield networks (Hopfield, 1982). After, we will review Dense Associative Memory, which generalizes on Hopfield networks and improves their capacity, with the loss of Hebbian weight updates. We will then review holographic associative memory models, which are similarly local (INSERT RESEARCH INTO HOLOGRAPHIC MEMORY HERE).

2 Background

> "Men make their own history, but they do not make it as they please; they do not make it under self-selected circumstances, but under circumstances existing already, given and transmitted from the past. The tradition of all dead generations weighs like a nightmare on the brains of the living." (Marx, 1852)

Dense Associative Memories build on a vast wealth of previous research, and while the desire for a simple and intuitive network which mirrors its qualities is straightforward, understanding how we get to this point requires an explanation of the thinking that got to where we are now. In this section we will detail the background information necessary for understanding Dense Associative Memories. To begin, we will discuss traditional Associative Memory models in subsection 2.1; next, we will discuss Correlation Matrix Associative Memories in subsection 2.2, which are the framework behind Hopfield networks, discussed in subsection 2.3. We will then discuss the theoretical capacity of Hopfield networks in subsection 2.4. Next we will discuss the theory behind Dense Associative Memories, and the energy-based framework developed behind them in subsection 2.5 and subsection 2.7. We will discuss their capacity as well in subsection 2.6. After, we will discuss local Hebbian learning, the theory behind the simplicity of Hopfield networks in subsection 2.8 and **??**. Finally, we will discuss related associative memory research: Holographic Associative Memories in **??**, **??**, and **??**. Finally,

2.1 Associative Memories

Associative Memory (AM) models are a family of architectures designed to learn to associate tuples of patterns (x, y) such that one can recover y from partially distorted, masked, or noisy forms of x. In order to generalize here, the following formal definition suffices:

Definition 2.1.1 (Associative Memories; auto-association; hetero-association)**.** An *associative memory* is a 3-tuple $\langle f, A, C\rangle$ obeying the following properties:

(i) The *address* matrix A is an $N \times D$ matrix of patterns we wish to learn as *cues* or *queries*;

(ii) The *content* matrix C is an $N \times M$ matrix of patterns we wish to learn as *responses*, or associate with each a_i, $i = 1, 2, \ldots, N$; and,

(iii) The *recall* function $f_{A,C} : \mathbb{R}^D \to \mathbb{R}^M$ maps D-dimensional query or cue patterns x to memorized M-dimensional patterns y.

An associative memory $\mathcal{M}$ is said to be *auto-associative* iff the address matrix is equivalent to the content matrix, i.e. $A = C$. It is sufficient, then, to only specify a recall function and address matrix for auto-associative memories. Associative memories are said to be *hetero-associative* otherwise.

Another typical property of AMs is *content-addressability*, which describes the task of recovering a stored content based on a partial subset of the information with it's associated query (Haykin, 2009; McClelland et al., 1986). This is in contrast to other memory models, for example in the C programming language (INSERT K&R citation here), where data is accessed in memory by arbitrary addresses unrelated to the data itself.

AMs also operate with a distinction between *recall* and *learning*. In recall, we apply the function $f_{A,C}$ to some D-dimensional pattern x to recover its associated content $\hat{y}$. This can be done by a single-pass of x through a neural network implementing $f_{A,C}$, or by asynchronous energy minimization (see, subsection 2.3). In the learning phase, the network adjusts $f_{A,C}$ via a learning algorithm to minimize the difference between the estimated content associated with a query, $\hat{y}$, and the desired associated content, y. This can be done via any learning algorithm that we choose.

AMs are not able to store an infinite amount of patterns. There is a critical capacity $N^{\max}$ for AMs (being the first dimension of the address and content matrices) at which recall can no longer adequately produce a $\hat{y}$ which corresponds with the desired associated pattern y. Critical capacity is influenced by the kind of data being learned. For example, a real-world dataset like MNIST (LeCun et al., 2010) contains images which are highly correlated with one another.

High correlation between stored patterns requires effective AMs to somehow be sensitive to fine-grain differences, which will be discussed in subsection 2.5 and subsection 2.7.

2.2 Correlation Matrix AMs

Correlation Matrix AMs are a class of auto-associative AMs which learn to associate bipolar patterns via a correlation matrix of desired patterns with themselves, originally presented in (Amari, 1972; Kohonen, 1988; Nakano, 1972):

Definition 2.2.1 (Correlation Matrix Associative Memory)**.** A *Correlation Matrix* associative memory is an auto-associative memory $\langle f, \Xi \rangle$, where:

(a) The $N \times D$ address matrix Ξ contains N patterns ξ_i of D-dimension; and,

(b) the recall function is:

$$f_\Xi(x) = g\left(\Xi^\top \Xi x\right),$$

and g is some function.

Correlation Matrix AMs are so-called as they recall values via the correlation matrix (in the simplest case) $W = \Xi^\top \Xi$. For Kohonen (1988), g is just the identify function $g(x) = x$. In Amari (1972) and Hopfield (1982), g is the *signum* function:

$$\operatorname{sgn}[n] = \begin{cases} -1, & \text{if } n < 0, \\ 0, & \text{if } n = 0, \\ 1, & \text{if } n > 0. \end{cases}$$

For a difference between the two g functions, see Figure 2.2. With a non-linear g function, have a clarified result which is forced to chose between the bipolar values $\{-1, 1\}$. Whereas, with $g(x) = x$, it becomes clear that the reconstructed value is a linear combination of the stored patterns weighted by their dot product with the query. To see how this is the case, suppose that we have pattern matrix $\Xi = [\xi_1, \xi_2, \ldots, \xi_N]$, and we wish to recall a query pattern σ. Recall, then, is:

$$\begin{aligned} f_\Xi(\sigma) &= \Xi^\top \Xi \sigma \\ &= (\xi_1 \xi_1^\top + \xi_2 \xi_2^\top + \cdots + \xi_N \xi_N^\top)\sigma \\ &= (\xi_1 \xi_1^\top \sigma) + (\xi_2 \xi_2^\top \sigma) + \cdots + (\xi_N \xi_N^\top \sigma) \\ &= \sum_{i=1}^{N} \xi_i \left(\xi_i^\top \sigma\right). \end{aligned} \tag{1}$$

The structure of Equation 1 also shows us how one could simply learn new representations over a period of time $T = N$. To begin, at time $t = 0$, we set the weight matrix W to zeros:

$$W^{(0)} \leftarrow [0]_{D \times D}. \tag{2}$$

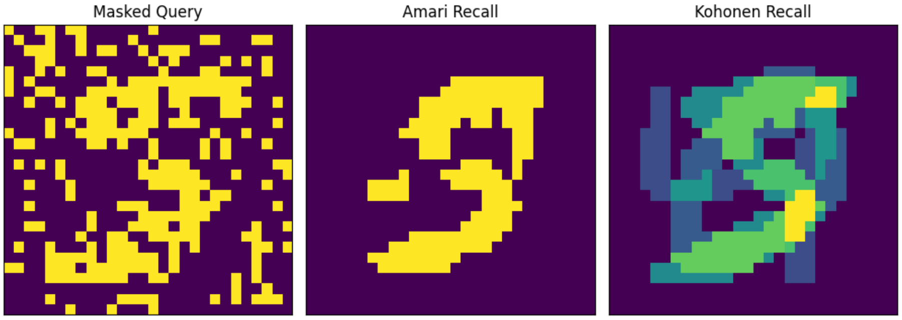

Figure 1: Left-to-right: Masked query presented to the networks. Single-pass recall with $g(x) = \mathrm{sgn}[x]$ ("Amari Recall"). Single-pass recall with $g(x) = x$ ("Kohonen Recall").

At each new time step $t + 1$, we update the weights with the corresponding pattern we wish to store:

$$W^{(t+1)} \leftarrow W^{(t)} + \xi_{(t+1)}\xi_{(t+1)}^{\top}. \tag{3}$$

This definition gives us the following *Activity Product Rule* (Haykin, 2009),

$$\Delta W = \xi_i \xi_i^{\top}. \tag{4}$$

We will discuss the Activity Product Rule and other Hebbian learning rules in subsection 2.8.

2.3 Hopfield Networks

The main contribution of the work of Hopfield (1982) is not the correlation-matrix structure of the associative memory. We have already discussed the idea of a single layer model with full, lateral connections above in subsection 2.2. Rather, the main contribution of Hopfield was the introduction of the notion of *energy*, and *energy descent*. Hopfield networks are characterized by an *energy function*, $E(\sigma^{(t)})$ which describes the current state of the network. If the energy function is a Lyapunov function (INSERT DISCUSSION OF LYAPUNOV FUNCTIONS HERE), and is well-designed, then the local minima of the range of the function will correspond up to a certain critical capacity with patterns that we wish the network to store.

More informally, the intuition is that we want to create an "energy landscape" via the energy function with dips and divots corresponding to patterns that we wish to store. "Energy descent" is so called because we start with an initial state of the network, and continuously lower the energy until it can be no longer lowered. In the same way that a ball set on the incline of a hill will fall till it reaches the lowest point, so too does an energy-based associative memory perform recall.

Definition 2.3.1 (Hopfield network)**.** A *Hopfield network* is a single-layer neural network which as D computational units with full, lateral connections. The state of the D computational units. Let the $N \times D$ matrix $\Xi = [\xi_1, \xi_2, \ldots, \xi_N]$ be the pattern matrix. The state of the D computational units, $\sigma^{(t)}$, depends on an *energy function*:

$$E(\sigma) = -\frac{1}{2}\sum_{i=1}^{N}(\xi_i \cdot \sigma)^2; \tag{5}$$

as,

$$\sigma_i^{(t+1)} = \arg\min_{b \in \{-1,1\}} \{E(\sigma_i^-), E(\sigma_i^+)\} \tag{6}$$

with $\sigma_i^{\pm}$ being $\sigma^{(t)}$ with the i'th element multiplied by ±1 (b).

We can in fact directly demonstrate that Hopfield networks are approximately equivalent to Correlation Matrix AMs. This proof is essential for understanding later developments in the theory of AMs in the same family as Hopfield networks, and will be discussed in subsection 2.5.

Theorem 2.3.1 (Hopfield networks are Correlation Matrix AMs)**.** *Let us have D computational units σ and $N \times D$ pattern matrix. Then, updating all elements σ_i, $i = 1, 2, \ldots D$ of the state vector by flipping their bit and keeping it if it lowers the energy is approximately equivalent to the single-pass update rule of correlation matrix AMs.*

Derivation and intuition come from Krotov et al. (2025). Recall that by Definition 2.3.1 the one-step update rule of a single random bit is:

$$\sigma_i^{(t+1)} = \arg\min_{b \in \{-1,1\}} \{E(\sigma_i^-), E(\sigma_i^+)\} \tag{7}$$

Let $F_n(\cdot) = \frac{1}{n}(\cdot)^n$, so $F_2 = \frac{1}{2}(\cdot)^2$. With F_2, we can reformulate the energy function of a Hopfield network as:

$$E(\sigma) = \sum_{i=1}^{N} F_2(\xi_i \cdot \sigma). \tag{8}$$

By Krotov et al. (2025), we can make a single pass update rule:

$$\forall i \in \{1, \ldots, D\},\ \sigma_i^{(t+1)} \approx \operatorname{sgn}\left[\sum_{j=1}^{N} \xi_{ji} F_2'\left(\xi_j \cdot \sigma^{(t)}\right)\right], \tag{9}$$

or,

$$\sigma^{(t+1)} = f_\Xi(\sigma^{(t)}) = \operatorname{sgn}\left[\Xi^T \Xi \sigma^{(t)}\right], \tag{10}$$

with $\Xi = [\xi_1, \xi_2, \ldots, \xi_N]$. This is just the single-pass update rule from Definition 2.2.1 using the *signum* function for g.

2.4 Hopfield Network Capacity

While the simplicity and appeal of Hopfield networks is clear, there is a problem for their plausibility as a computational model of the mechanisms for memory in the brain. Namely, their limited capacity, with the famous result from Hopfield (1982) that the critical capacity of patterns that the network can reliably retrieve is around 15 per cent of the number of computational units of the network. Of relevance for our discussion is how we can systematically arrive at a critical capacity estimate for any associative memory. Therefore, here we will see how one can arrive at the capacity result for Hopfield networks.

The critical capacity of a Hopfield network can be arrived at by deriving an upper bound on the number of patterns that the network can store while still having that any pattern stored in the network will be recalled with minimal errors (Demircigil et al., 2017; Krotov & Hopfield, 2016; Krotov et al., 2025). More formally, let $\Xi = [\xi_1, \dots, \xi_N]$ be the set of D-dimensional desired patterns that we wish for the network to store, and let σ be the D-dimensional vector representing the current state of the network at time t. Then, we initialize $\sigma^{(t=0)}$ to some pattern ξ_i, and let the network evolve. If $\sigma^{(t+1)}$ is roughly equivalent to $\sigma^{(t=0)}$ (w.r.t. some distance function $d(\sigma^{(t+1)}, \sigma^{(t=0)}) > \epsilon$, where ϵ is a constant) then the network is at an energy equilibrium, i.e., a desired stored pattern.

2.5 Dense Associative Memories

Hopfield networks are certainly simple and intuitive. But their strength is also their weakness: their simplicity leads to low storage capacity, as we have seen above. This problem, as well as convergent theoretical work in the study of associative memories unrelated to Hopfield networks, found an interesting solution: applying a non-linear function to the dot product scores between the desired stored patterns x_i and the query pattern σ.

Early models that used this technique are: Minerva2 (Hintzman, 1984), which computes the cosine similarity between the query pattern and stored patterns, and then sums the rows of the element-wise product of these weighted similarity scores and our desired patterns. Similar work investigated energy functions with tensor-based interaction terms (Chen et al., 1986; Psaltis et al., 1988), with a unifying approach found in Kelly et al. (2017).

A general theory can be found in Krotov and Hopfield (2016), later expanded in Demircigil et al. (2017), and to arbitrarily sized neural networks in Krotov (2021).

Definition 2.5.1 (Dense Associative Memory). *Dense Associative Memory* is a generalization of Hopfield networks, with $N \times D$ pattern matrix $\Xi = [\xi_1, \xi_2, \dots, \xi_N]$, D-dimensional state vector $\sigma^{(t)}$, and an energy function of the form:

$$E(\sigma) = -\sum_{i=1}^{N} F_n\left(\xi_i \cdot \sigma\right), \tag{11}$$

where F_n is a *polynomial* function $F_n(\cdot) = \frac{1}{n}(\cdot)^n$ (of the same form as Equation 8).

The form of the polynomial function F_n is non-trivial, as it relates to the single-pass rule, by the fact used in Equation 9. Namely, the *single-pass* rule uses the derivative of F_n, F_n', as its non-linear function. From this definition immediately follows the next two corollaries.

Corollary 2.5.1. *Hopfield networks of the form given in Definition 2.3.1 are Dense Associative Memories, with polynomial function F_2.*

To see why this is the case, note that $\frac{d}{dx}F_2' = \frac{1}{2}(2)(x)^{2-1} = x$.

Corollary 2.5.2. *The memory model* Minerva2 *from Hintzman (1984) is approximately a Dense Associative Memory with polynomial function F_4.*

With polynomial function F_4, the energy function becomes:

$$E(\sigma) = -\frac{1}{4}\sum_{i=1}^{N}(\xi_i \cdot \sigma)^4. \tag{12}$$

By Krotov et al. (2025), the update-rule for $\sigma^{(t+1)}$ is (using the same move as in Equation 9):

$$\sigma_i^{(t+1)} = \text{sgn}\left[\sum_{j=1}^{N} \xi_{ji}(\xi_j \cdot \sigma^{(t)})^3\right]. \tag{13}$$

Compare the transmission function of Minerva2:

$$\sigma^{(t+1)} = \text{sgn}\left[\sum_{i=1}^{N} \xi_i \left(\frac{\xi_i^\top \sigma^{(t)}}{\|\xi_i\|\|\sigma^{(t)}\|}\right)^3\right]. \tag{14}$$

If every ξ_i, $i = 1, 2, \ldots, N$ and $\sigma^{(t)}$ are vectors of magnitude 1, then Equation 13 and Equation 14 are directly equivalent.

The insights of Dense Associative Memories are not limited to binary vectors, and have been shown to apply to multi-head attention in transformer networks (Ramsauer et al., 2021; Vaswani et al., 2023).

Theorem 2.5.1. *Any DAM with polynomial function F_n, with $n > 2$, is not Hebbian.*

We say that an architecture is Hebbian if the connections between every layer x_i and x_j, $i \neq 0$ and $j = i + 1$, is entirely determined by the prior activity of x_i, x_j, and the previous connection weights between them. But, Krotov and Hopfield (2021) notes that $n \geq 3$ requires many-body neuron interactions, which are non-Hebbian.

One solution, to be discussed in subsection 2.9, is Neuron-astrocyte memory (Kozachkov et al., 2024). Related work (Ramsauer et al., 2021)

2.6 Dense Associative Memory Capacity

Theorem 2.6.1. *The critical capacity, N^{max} of any DAM with polynomial function F_n scales proportionally to D^{n-1},*

$$N^{max} \propto D^{n-1}.$$

See Demircigil et al. (2017) and Krotov and Hopfield (2016).

Previous work talks about the number of connections affecting the number of memories that are capable of being stored (Little & Shaw, 1978). More concrete proof here, reproduce from (Demircigil et al., 2017; Krotov & Hopfield, 2016). Signal-to-noise estimation.

2.7 Universal Hopfield Networks

Definition 2.7.1 (Universal Hopfield Networks)**.** *Universal Hopfield Networks* (UHNs) are a general theory of single-pass associative memories, where the recall function f can be broken into three parts:

$$f_{A,C}(\sigma) = \underbrace{C^T}_{\text{Projection}} \left(\underbrace{\text{sep}}_{\text{Separation function}} \left[\underbrace{\text{sim}(A, \sigma)}_{\text{Similarity function}} \right] \right).$$

Namely:

(1) A *similarity function*, that computes the relevant distance between stored patterns and query patterns;

(2) A *separation function*, which is typically non-linear, which maximizes high similarity values and minimizes low similarity scores; and,

(3) A *projection function*, which generates vectors in the row space of the content matrix, formed by the basis vectors of the row space of the column matrix weighted by their respective separated similarity scores.

While investigation into DAMs has revealed that their capacity scales with the F_n, with F'_n being their separation function; informal experimentation reveals that another important way of increasing capacity is the similarity function of the associative memory (Hu et al., 2024; Millidge et al., 2022).

2.8 Hebbian and Local Learning Rules

Definition 2.8.1 (Hebbian update)**.** A *Hebbian update rule* or *Hebbian learning* rule is an update rule which is entirely local. By *local weight update*, we mean that the weights connecting layers x_i and x_j, $j = i + 1$, are defined in terms of: the activity of x_i, the activity of $x_j = g(x_i)$, and the weights w_{ij}:

$$\Delta w_{ij} = f(w_{ij}; x_i, x_j). \tag{15}$$

Definition 2.8.2 (Expanded Hebbian update rule)**.** The *expanded* Hebbian learning rule is the Taylor expansion of the weight update rule about $(0,0)$ (Gerstner & Kistler, 2002):

$$\begin{aligned} \Delta w_{ij} &= f(w_{ij}; x_i, x_j) \\ &= f(w_{ij}; 0, 0) + \frac{\partial f}{\partial x_i}\big|_{(0,0)} x_i + \frac{\partial f}{\partial x_j}\big|_{(0,0)} x_j \\ &+ \frac{1}{2}\frac{\partial^2 f}{\partial v_i^2}\big|_{(0,0)} x_i^2 + \frac{1}{2}\frac{\partial^2 f}{\partial v_j^2}\big|_{(0,0)} x_j^2 \\ &+ \frac{\partial^2 f}{\partial x_i \partial x_j}\big|_{(0,0)} x_i x_j + \mathcal{O}(x). \end{aligned}$$

We denote the Taylor coefficients by $c_n^{\{\text{corr, pre, post}\}}$, with n denoting the order of the coefficient, corr denoting the coefficient weighting the correlation between the pre- and post-synaptic neural layers, and pre and post to denote the coefficients of the pre- and post-synaptic layers themselves (respectively). This gets us:

$$\begin{aligned} \Delta w_{ij} = f(w_{ij}; x_i, x_j) &= c_0(w_{ij}) + c_1^{\text{pre}}(w_{ij}) x_j + c_1^{\text{post}} x_i \\ &+ c_2^{\text{pre}}(w_{ij}) x_j^2 + c_2^{\text{post}}(w_{ij}) x_i^2 + c_2^{\text{corr}}(w_{ij}) x_i x_j \\ &+ \mathcal{O}(x). \end{aligned}$$

Corollary 2.8.1 (Activity Product Rule)**.** *We can derive the Activity Product Rule in Equation 4 by setting c_2^{corr} to some constant, and letting all other coefficients be zero, getting us an update rule:*

$$\Delta w_{ij} = c_2^{corr}(w_{ij}) x_i x_j.$$

Typically, we denote $c_2^{corr}(w_{ij})$ by η, the learning rate.

Corollary 2.8.2 (Oja's Rule)**.** *Oja's rule (Oja, 1982) is a competitive Hebbian learning algorithm where the weights are naturally normalized. We can derive it from Definition 2.8.2 by setting $c_2^{corr} = \eta_0 > 0$, and $c_2^{post} = -\eta_0 w_{ij}$, with all other coefficients set to 0:*

$$\begin{aligned} f(w_{ij}; x_i, x_j) &= \eta_0 x_i x_j - \eta_0 w_{ij} x_i^2 \\ &= \eta_0 x_i (x_j - w_{ij} x_i). \end{aligned}$$

2.9 Related Work

(a) Neuron-astrocyte networks

(b) DAM in feature space

(c) Kernel-based DAMS

(d) Predictive Coding Networks

3 Work Plan

Completing this project requires two tasks which are interrelated but practically independent in their execution. The first is the theoretical work involved in finding a general theory of the capacity of Hebbian alternatives to Dense Associative Memories. The second is large-scale experimentation on real-world datasets We will discuss the theoretical task in subsection 3.1, the experimental work in subsection 3.2.

3.1 Theoretical Work

Following Definition 2.8.2, we can generate a range of different Hebbian learning algorithms for a single-layer network with full lateral connections. All of which are entirely local, and therefore biologically plausible. There are established methods for deriving the theoretical capacity of associative memories, as we have seen in subsection 2.4 and subsection 2.6. Therefore, the goal in this part of the section is to come up with a general theorem about the critical capacity of associative memories with Hebbian update rules. Importantly, the goal here is to provide a parameterized theorem: *how do the choices of coefficients in the Expanded Hebbian learning rule each affect the critical capacity.*

There are two ways of approaching this goal: the first is to ignore previous Hebbian learning rule research and attack the problem through permutation of coefficient values. While this approach would be time-consuming, it would be extensive, and maybe lead to insight from fresh eyes on the problem. The second way of approaching the problem is to start from extant Hebbian learning algorithms. Then, we would prove (or find a proof) for each a critical capacity as a learning rule for an associative memory. This would be less time-consuming, and provide the project with clearer milestones and goals. For this approach to be novel, it is not sufficient to just reiterate the proofs of each update rule's critical capacity. Rather, after finding or writing each proof, it would be helpful to create a general theorem for all of them based on common insights.

3.2 Experimental Work

The second part of the work is experimentation with different Hebbian learning based associative memories, and comparing their performance to contemporary models of associative memory. The performance metric by which to compare each model highly depends on the task. The conventional task for introducing associative memory models is to do a qualitative evaluation of their ability to do pattern completion from partially obscured or noisy queries. While this approach does provide intuitive results, it is hardly an objective measure beyond broad statements like: the results of this associative memory “look” better than the results of some other.

The second approach for evaluating the performance of associative memories is to use them for some task as part of a larger network. For example, we can evaluate the associative memory in a classification task as a part of a larger network where we input into the network partial or obscured representations of our domain of interest, and classify the results of the recall. This latter approach seems to be a better fit for the scientific evaluation of differing associative memories.

4 Timeline

"The war will be over by Christmas!" (Many)

(a) Theoretical investigation by December, paper to follow

(b) Option to either split the experimentation from the theory for confirmation as a separate paper, or as a single paper

(c) Fitting in Hebbian AMs into Transformer networks; or something similar; do they hold water to DAMs in sequence prediction/image generation?

(d) Do we see similar memorization → generalization as DAMs exhibit?

References

Amari, S.-I. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers*, *C-21*(11), 1197–1206. https://doi.org/10.1109/T-C.1972.223477

Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1985). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, *55*(14), 1530–1533. https://doi.org/10.1103/PhysRevLett.55.1530

Chen, H. H., Lee, Y. C., Sun, G. Z., Lee, H. Y., Maxwell, T., & Giles, C. L. (1986). High order correlation model for associative memory [ISSN: 0094243X]. *AIP Conference Proceedings*, *151*, 86–99. https://doi.org/10.1063/1.36224

Demircigil, M., Heusel, J., Löwe, M., Upgang, S., & Vermet, F. (2017). On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, *168*(2), 288–299. https://doi.org/10.1007/s10955-017-1806-y

Gerstner, W., & Kistler, W. M. (2002). Mathematical formulations of hebbian learning. *Biological Cybernetics*, *87*(5), 404–415. https://doi.org/10.1007/s00422-002-0353-y

Haykin, S. S. (2009). *Neural networks and learning machines* (3. ed). Prentice-Hall.

Hintzman, D. L. (1984). MINERVA 2: A simulation model of human memory. *Behavior Research Methods, Instruments, &amp Computers*, *16*(2), 96–101. https://doi.org/10.3758/BF03202365

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*, 2554–2558.

Hu, J. Y.-C., Wu, D., & Liu, H. (2024, October 31). Provably optimal memory capacity for modern hopfield models: Transformer-compatible dense associative memories as spherical codes. https://doi.org/10.48550/arXiv.2410.23126

Kelly, M. A., Mewhort, D., & West, R. L. (2017). The memory tesseract: Mathematical equivalence between composite and separate storage memory models. *Journal of Mathematical Psychology*, *77*, 142–155. https://doi.org/10.1016/j.jmp.2016.10.006

Kohonen, T. (1988). Correlation matrix memories. In J. A. Anderson & E. Rosenfeld (Eds.), *Neurocomputing, volume 1* (pp. 174–180). The MIT Press. https://doi.org/10.7551/mitpress/4943.003.0075

Kozachkov, L., Slotine, J.-J., & Krotov, D. (2024, July 23). Neuron-astrocyte associative memory. https://doi.org/10.48550/arXiv.2311.08135

Krotov, D. (2021, July 14). Hierarchical associative memory. https://doi.org/10.48550/arXiv.2107.06446

Krotov, D., Hoover, B., Ram, P., & Pham, B. (2025, July 8). Modern methods in associative memory. https://doi.org/10.48550/arXiv.2507.06211

Krotov, D., & Hopfield, J. (2021, April 27). Large associative memory problem in neurobiology and machine learning. https://doi.org/10.48550/arXiv.2008.06996

Krotov, D., & Hopfield, J. J. (2016, September 27). Dense associative memory for pattern recognition. https://doi.org/10.48550/arXiv.1606.01164

LeCun, Y., Cortes, C., & Burges, C. (2010). MNIST handwritten digit database. *ATT Labs*, *2*. http://yann.lecun.com/exdb/mnist

Little, W. (1974). The existence of persistent states in the brain. *Mathematical Biosciences*, *19*(1), 101–120. https://doi.org/https://doi.org/10.1016/0025-5564(74)90031-5.

Little, W., & Shaw, G. L. (1978). Analytic study of the memory storage capacity of a neural network. *Mathematical Biosciences*, *39*(3), 281–290. https://doi.org/10.1016/0025-5564(78)90058-5

Marr, D. (1971). Simple memory: A theory for archicortex. *Philosophical Transactions of the Royal Society of London*, *262*(841), 23–81.

Marx, K. (1852). *The eighteenth brumaire of louis bonaparte*. Die Revolution. Retrieved September 23, 2025, from https://www.marxists.org/archive/marx/works/1852/18th-brumaire/

McAlister, H., Robins, A., & Szymanski, L. (2025, March 4). Sequential learning in the dense associative memory. https://doi.org/10.48550/arXiv.2409.15729

McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986, July 17). The appeal of parallel distributed processing. In *Parallel distributed processing: Explorations in the microstructure of cognition: Foundations* (pp. 3–44, Vols. 2, Vol. 1). The MIT Press. Retrieved August 28, 2025, from https://doi.org/10.7551/mitpress/5236.001.0001

Millidge, B., Seth, A., & Buckley, C. L. (2022, July 12). Predictive coding: A theoretical and experimental review. https://doi.org/10.48550/arXiv.2107.12979

Nakano, K. (1972). Associatron-a model of associative memory. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-2*(3), 380–388. https://doi.org/10.1109/TSMC.1972.4309133

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*, 267–273.

Pham, B., Raya, G., Negri, M., Zaki, M. J., Ambrogioni, L., & Krotov, D. (2025, June 20). Memorization to generalization: Emergence of diffusion models from associative memory. https://doi.org/10.48550/arXiv.2505.21777

Psaltis, D., Park, C. H., & Hong, J. (1988). Higher order associative memories and their optical implementations. *Neural Networks*, *1*(2), 149–163. https://doi.org/10.1016/0893-6080(88)90017-2

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., & Hochreiter, S. (2021, April 28). Hopfield networks is all you need. https://doi.org/10.48550/arXiv.2008.02217

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986, July 17). A general framework for parallel distributed processing. In *Parallel distributed processing: Explorations in the microstructure of cognition: Foundations* (pp. 45–76, Vols. 2, Vol. 1). The MIT Press. Retrieved August 28, 2025, from https://doi.org/10.7551/mitpress/5236.001.0001

Stanley, J. C. (1976). Simulation studies of a temporal sequence memory model. *Biological Cybernetics*, *24*(3), 121–137. https://doi.org/10.1007/BF00364115

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023, August 2). Attention is all you need. https://doi.org/10.48550/arXiv.1706.03762