Investigating Hebbian Alternatives to Dense Associative Memory

Connor Hanley

October 15, 2025

Abstract

Dense Associative Memories generalize traditional Hopfield networks while providing a substantially increased capacity. To do this, they show that by increasing the separation of similarity scores between query patterns and stored patterns. The cost of increased capacity is simplicity and biological plausibility. We propose to investigate the capacity characteristics of all Hebbian associative memories, to see if there exists any alternative to Dense Associative Memories which maintains the simplicity and appeal of Hopfield networks.

1 Introduction

Humans are able to recognize and retrieve patterns of data using distorted, noisy, and partial patterns (Rumelhart, Hinton, & McClelland, 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). 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).

The foundational model for modern associative memory research is the *Hopfield network* (J. J. Hopfield, 1984; J. Hopfield, 1982). Hopfield networks are single-layer neural networks with full, lateral connections. This means that each artificial neuron in the network is connected with every other neuron, except itself. Patterns that we wish to recall from the network are stored in the connection weights between every other unit. Hopfield networks learn new patterns to recall through a simple, biologically plausible learning rule called the *Activity Product Rule* (Haykin, 2009), which is a kind of Hebbian update rule (Hebb, 1949). The

appeal of Hebbian update rules is that they are simple, in that they are totally defined by local interactions between layers in neural networks, and they are biologically plausible, having been observed in interactions between biological neurons (Bi & Poo, 1998; Markram et al., 1997; Rolls, 2013).

In spite of their simplicity and biological plausibility, Hopfield networks are inherently flawed. The number of patterns that Hopfield networks can store is pitifully small: with estimates in the range of 14% to 15% of the number of neurons in the network (Amit et al., 1987; J. Hopfield, 1982). In order to remedy the gap between the simplicity and plausibility of Hopfield networks and their child Dense Associative Memory models, we will theoretically derive and empirically test the storage capacities of all Hebbian update rules when used in single-layer neural networks with full lateral connections.

Discovering the limits of Hebbian alternatives to Dense Associative Memories is not only desirable because of simplicity and biological plausibility (while those are both laudable goals). For example, the study of Vector-Symbolic Architectures, used for representing high-level cognitive-tasks (Gayler, 1998; Kelly et al., 2013; Plate, 1995; Smolensky, 1990), requires an auto-associative memory "clean-up memory". This means that the expressive capabilities of Vector-Symbolic Architectures are affected by the capacity of the associative memory used.

In the following we will discuss the problem in more depth. To do this, we will first lay out the theoretical motivation: beginning with an introduction to the literature in section 2, covering associative memories in subsection 2.1, Hopfield networks in subsection 2.2, their generalization with Dense Associative Memories in subsection 2.3, and Hebbian learning and learning rules in subsection 2.4. We will restate the problem of Hebbian Dense Associative Memories in section 3, and provide a work plan in subsection 3.1. Finally, we will discuss the timeline for completing this project in section 4.

2 Background

In order to understand the need for a Hebbian alternative to Dense Associative Memories, we must first familiarize ourselves with the literature up to this point. Associative Memory research has a wealth of literature and alternatives. In order to limit the scope of this research, we limit ourselves only to the family of models based on Hopfield networks (J. Hopfield, 1982).

2.1 Associative Memories

An associative memory, in particular an *auto*-associative memory, is a general structure that can be implemented by many mathematical and computational objects. Broadly speaking, an associative memory is a tuple of a set of objects $\mathcal{X}$, called the set of *traces* or *stored patterns*, and a (typically learned) identity map

over the set of stored patterns, $T : \mathcal{X} \to \mathcal{X}$. More formally,

Definition 2.1.1 (Associative memories)**.** An (auto) *associative memory* is a tuple $\langle \mathcal{X}, T \rangle$, where

(a) $\mathcal{X}$ is a set of objects, called the *pattern* set, or set of *traces*, typically high-dimensional vectors; and,

(b) The *recall function*, T, maps from the set of traces back to the set of traces, such that $T(x) \approx x$, for all $x \in \mathcal{X}$, and $T(\overline{x}) \approx x$, where $\overline{x}$ is a perturbed, masked, or degraded form of $x \in \mathcal{X}$.

We will be assuming throughout that the pattern set is the set of rows of the *pattern matrix* $\Xi = [\xi^1, \xi^2, \ldots, \xi^N]$, where each *stored pattern* $\xi^i \in \{-1, 1\}^D$. It is sufficient to specify the pattern matrix, or the sequence of patterns, for an associative memory, and it should be assumed that the pattern set is just composed of the rows of the pattern matrix. Associative memories are *content-addressable*, as their recall function is able to recover desired associated patterns from partial information (Haykin, 2009; McClelland et al., 1986). This is in contrast to memory models which recall information based on arbitrary associations, like indices into memory locations.

While it is not essential to the most general definition of associative memories, we usually desire that the recall function be learned. Furthermore, associative memories have a maximum capacity of traces that they can store.

Definition 2.1.2 (Retrievability; Capacity)**.** Following Bao et al. (2022), let us have an associative memory with patterns $\xi^1, \xi^2, \ldots, \xi^N$, where each element ξ^i_j is -1 or 1 with equal likelihood. For any $\delta < 0.5$, we define the δ-perturbation of ξ^i, $\bar{\xi}^i$, as the D-dimensional vector ξ^i with each element flipped with a likelihood of δ. Then, we say that the set $\xi^1, \xi^2, \ldots, \xi^N$ is (δ, ε)-retrievable if for every ξ^i, $i = 1, 2, \ldots, N$ it is such that:

$$\mathbb{P}(T(\bar{\xi}^i) \neq \xi^i) < \varepsilon. \tag{1}$$

The *capacity* C of the associative memory is the maximum cardinality of the pattern set such that all patterns are (δ, ϵ)-retrievable.

In Krotov and Hopfield (2016), capacity is denoted by $N^{\max}$, where N is the first dimension of the pattern matrix. Associative memory capacity is affected by the (1) kind of data being stored, i.e. if the data has a high pairwise correlation, which leads to "cross-talk" (Kohonen, 1988), (2) the dimensionality of the data, and (3) the implementation of the recall function T. For example, if the associative memory is a neural network with tensor weights (Kelly et al., 2017), then the capacity scales with the number of the weights (Little & Shaw, 1978) (expanded further in subsection 2.3).

Before Hopfield networks, there were Correlation Matrix associative memories. So-called, because they relied on dot-product correlations for recall.

Example 2.1.1 (Correlation Matrix associative memory)**.** *Let us have a pattern matrix* $\Xi = [\xi^1, \xi^2, \ldots, \xi^N]$ *of vectors sampled from* $\{1, -1\}^D$*. The Correlation Matrix associative memory recall function is of the form:*

$$\begin{aligned} T(x) &= g\left(\sum_{\mu=1}^{N} \xi^\mu \left(\sum_{i=1}^{D} \xi_i^\mu x_i\right)\right) \\ &= g\left(\Xi^\top \Xi x\right), \end{aligned} \tag{2}$$

where g *is an "activation function", typically signum for binary vectors.*

Example 2.1.2. *Like above, let us have a pattern matrix* $\Xi = [\xi^1, \xi^2, \ldots, \xi^N]$ *of vectors sampled from* $\{1, -1\}^D$*. Let our recall function* T *be:*

$$T(x) = \xi^i, \text{ where } i = \arg\max_{i \in [1,N]} [sim(\xi^i, x)], \tag{3}$$

where sim is some similarity function (Kelly et al., 2017), e.g. the cosine similarity or Hamming distance.

Example 2.1.3 (Minerva2)**.** *Minerva2 (Hintzman, 1984) is an associative memory used in cognitive science which is about as old as Hopfield networks. As per usual, we have binary patterns* $\Xi = [\xi^1, \xi^2, \ldots, \xi^N]$*. The recall function is:*

$$T(x) = sgn\left[\sum_{\mu=1}^{N} \xi^\mu \left(\frac{\sum_{i=1}^{D} \xi_i^\mu x_i}{\|\xi^\mu\| \|x\|}\right)^3\right]. \tag{4}$$

2.2 Hopfield Networks

Hopfield networks (J. J. Hopfield, 1984; J. Hopfield, 1982) are an associative memory with deep connections to Example 2.1.1, except that they understand their recall function in terms of an *energy* function:

Definition 2.2.1 (Hopfield networks)**.** Hopfield networks are single-layer neural networks of D computational units with full, lateral connections. The state of the network is described by the state vector σ at time t, denoted by $\sigma^{(t)}$. The *energy* of the network, $E(\sigma^{(t)})$ is:

$$E(\sigma^{(t)}) = -\frac{1}{2} \sum_{\mu=1}^{N} \left(\sum_{i=1}^{D} \xi_i^\mu \sigma_i^{(t)}\right)^2. \tag{5}$$

Let $\sigma[i := b]$ be the vector σ, with the i'th component set to b. Then, update rule for the state of the network is:

$$\sigma_i^{(t+1)} = \text{sgn}\left[E(\sigma^{(t)}[i := -1]) - E(\sigma^{(t)}[i := 1])\right]. \tag{6}$$

After setting the initial state at time $t = 0$, the network continuously updates random elements of the

state vector depending upon whether the update lowers the energy of the network. The energy function, then, forms a "landscape", where local minima in the space (ideally) correspond with desired patterns to be stored.

While the use of the energy function seems at odds with our definition of associative memories, we can align the two by considering a single-pass variant of the energy-minimization. It can be shown that updating every element i of the state vector $\sigma^{(t)}$ in a single pass (synchronously) corresponds with the update rule (Krotov et al., 2025):

$$T(\sigma) = \text{sgn}\left[\sum_{\mu}^{N} \xi^{\mu}\left(\sum_{i=1}^{D} \xi_i^{\mu}\sigma_i\right)\right], \tag{7}$$

which is just the update rule of Correlation Matrix associative memories (see, Example 2.1.1).

Hopfield networks unfortunately have a pitifully low storage capacity: the critical capacity C being around 14% to 15% of the number of neurons in the model (Amit et al., 1987; J. Hopfield, 1982). As a computational model of human memory this will not do. Likewise, this is unsuitable for machine learnings tasks involving large datasets, e.g. large language models. Capacity problems led researchers to investigate how one can improve the capacity of Hopfield networks, while maintaining the simplicity and interpretability of energy minimization.

2.3 Dense Associative Memories

Dense Associative Memory models are a generalization of Hopfield networks that maintain energy minimization in recall, but have a dramatically increased storage capacity (Demircigil et al., 2017; Krotov & Hopfield, 2016). Instead of relying on full lateral connections between units in a single dimension, Dense Associative Memories either have multiple connections between computational units, or, increase the number of layers present in the model. Krotov and Hopfield (2021) notes that this is to be expected, given the information limits of single neurons.

Definition 2.3.1 (Dense Associative Memory)**.** Given binary patterns $\xi^1, \xi^2, \ldots, \xi^N$ sampled from $\{-1, 1\}^D$, a *Dense Associative Memory* is characterized by:

(i) A D-dimensional *state vector* at time t, $\sigma^{(t)}$

(ii) A *polynomial transmission function* F_n, where:

$$F_n(x) = \frac{1}{n}(x)^n; \tag{8}$$

(iii) An *energy function* of the state of the network at time t,

$$E(\sigma^{(t)}) = \sum_{\mu=1}^{N} F\left(\sum_{i=1}^{D} \xi_i^\mu \sigma_i^{(t)}\right), \tag{9}$$

(iv) An update equation for the state of the network, such that:

$$\sigma_i^{(t+1)} = \text{sgn}\left[E(\sigma^{(t)}[i := -1]) - E(\sigma^{(t)}[i := 1])\right]. \tag{10}$$

Like Hopfield networks, the state vector update rule is typically performed asynchronously. However, we can define a synchronous update rule which makes the Dense Associative Memory an Associative Memory with update rule:

$$T(\sigma) = \text{sgn}\left[\sum_{\mu=1}^{N} \xi^\mu F_n'\left(\sum_{i=1}^{D} \xi_i^\mu \sigma_i\right)\right], \tag{11}$$

where F_n' is the derivative of F_n. From this definition we immediately see how this framework generalizes Hopfield networks.

Example 2.3.1. *Hopfield networks in Definition 2.2.1 are Dense Associative Memories with polynomial transmission function F_2.*

Example 2.3.2. *Minerva2 (from Example 2.1.3) are roughly Dense Associative Networks, assuming that all patterns $\xi^1, \xi^2, \ldots, \xi^N$ and query patterns are normalized, and with polynomial transmission function F_4.*

Dense Associative Memories improve the storage capacity of Hopfield networks in that the capacity C of a Dense Associative Memory, with polynomial function F_n, and dimensionality of patterns D is:

$$C \propto D^{n-1} \tag{12}$$

(Bao et al., 2022; Demircigil et al., 2017; Krotov & Hopfield, 2016). The increased capacity comes from the "steeper" energy basins around minima corresponding to patterns stored in the network, which is brought about by the exponentiation of the dot product similarity between the stored patterns and the query pattern (for discussion, see Kelly et al. (2017)). Memory capacity for Dense Associative Memories can also be increased by making more domain sensitive similarity functions (Millidge et al., 2022), e.g. with kernel estimation (Hu et al., 2024; Wu et al., 2024).

In spite of the enormous gains in capacity using non-linear transmission functions, Dense Associative Memories lose the ability to be interpreted as simple one-layer networks with lateral connections and trained via Hebbian update rules (Krotov & Hopfield, 2021; McAlister et al., 2025). On the one hand, this is to their

benefit: a continuous variant of Dense Associative Memory called *Modern Hopfield Networks* has been shown to be equivalent to multi-head attention in Transformer neural networks (Ramsauer et al., 2021; Vaswani et al., 2023). Likewise, the framework behind Dense Associative Memories has been developed for modular neural network architectures (Krotov, 2021), and has been used to define an *Energy Based Transformer* (Hoover et al., 2023), as well as contribute to the interpretability of Diffusion Models (Pham et al., 2025).

But the cost is their simplicity and biological plausibility. Correlation matrix memories trained with Hebbian update rules mimic observed phenomena in biological neural networks; meaning that they perform computations which are in principle achievable by biological systems.

2.4 Hebbian Learning Rules

Hebbian learning rules are a family of weight update rules for both artificial and spiking neural networks which posit that the weights connecting two local layers are determined entirely by the activity of just those two layers (Hebb, 1949; Sejnowski & Tesauro, 1989). We can define Hebbian learning rules in the most general sense:

Definition 2.4.1 (Hebbian learning rule)**.** Consider a neural network with layers with N layers $x_1, x_2, \ldots, x_N$, and for each layer and the layer above (i and $i+1$, $i \neq N$), there is a *weight matrix* W_{ij}. Then, a Hebbian learning rule for this network is an update rule for layers i, j, where $j = i + 1$, such that:

$$\Delta W^{ij} = F(W^{ij}; x^i, x^j). \tag{13}$$

Hebbian learning rules are *entirely local*, as we can see from the requirement that the learning rule only affect layers i and $j = i + 1$. In their simplest formulation, Hebbian learning rules are not all error-correcting (McClelland, 2006), which is in contrast to popular learning algorithms like back-propagation of error (Rumelhart, Hinton, & Williams, 1986). However, there is a local learning paradigm, called Predictive Coding (for review see Spratling (2017)), which has implemented associative memories in deep neural networks (Salvatori et al., 2021)

The simplest method for training Hopfield networks (and Correlation Matrix associative memories) is to use the Activity Product Rule (Haykin, 2009).

Definition 2.4.2. The *Activity Product Rule* is a Hebbian learning rule defined for layers x^i and x^j, $j = i+1$, with weights W^{ij}, such that:

$$\Delta W^{ij} = \eta x^i (x^j)^\top, \tag{14}$$

where we call η the *learning rate*.

To train Hopfield networks using the Activity Product Rule, let us have patterns $\xi^1, \xi^2, \ldots, \xi^N$; a single D-dimensional layer σ, and full lateral weights between σ and itself called W. At training step $t = 0$, we initialize W to be all zeros:

$$W_{ij}^{(t=0)} = 0. \tag{15}$$

Then, for each new time step from $t = 1$ to $t = N$, we produce a new pattern to the weights for it to memorize, updating it with the Activity Product Rule:

$$W^{(t+1)} \leftarrow W^{(t)} + \eta \xi^{t+1} (\xi^{t+1})^\top. \tag{16}$$

Hopfield networks learn with the Activity Product Rule (Haykin, 2009). Talk about biological evidence for Hebbian learning rules. Stress simplicity. If $\eta = 1$, then we get a weight matrix W which is a linear combination of all the desired patterns we wish to store:

$$W^{(t=N)} = \sum_{\mu=1}^{N} \xi^\mu (\xi^\mu)^\top. \tag{17}$$

Weights trained in this way makes the synchronous recall function T for the Hopfield network a linear combination of the stored patterns weighted by their dot product similarity with the query pattern:

$$T(\sigma) = \text{sgn}\left[W^{(t=N)}\sigma\right] = \text{sgn}\left[\sum_{\mu=1}^{N} \xi^\mu (\xi^\mu)^\top \sigma\right]. \tag{18}$$

In principle, auto-associative memories could be trained with *other* Hebbian update rules, as long as the Hebbian update rule was additive and can be understood as reducing the energy of the network (according to some rule). More strictly, a learning rule is additive when, after training, the weights are of the form:

$$W^{(t_{\text{final}})} = \sum_{\mu=1}^{N} f(W; \xi^\mu (\xi^\mu)^\top). \tag{19}$$

This property is required in order to maintain the neural interpretation of recall, as post-multiplication distributes over element-wise addition. The second important property of the weight update rule is that it reduces some natural energy function for the update rule. For example, if we derive the energy rule through the relation from Krotov (2021).

Given that f in Definition 2.4.1 is a large class of functions, we will restrain ourselves to only functions of the following form:

Definition 2.4.3. 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:

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

This definition immediately gives us the following:

Example 2.4.1. *The Activity Product Rule (Definition 2.4.2) is just the expanded Hebbian learning rule with c_2^{corr} set to some constant, and all other constants set to* 0*.*

Example 2.4.2. *Oja's Rule (Oja, 1982) by setting $c_2^{corr} = \eta_0 > 0$, and $c_2^{post} = -\eta_0 w_{ij}$, with all other coefficients set to* 0*.*

3 Hebbian Dense Associative Memory

Instead of manipulating the similarity kernel to increase memory capacity (as in Hoover et al. (2024) and Hu et al. (2024)), can we use different Hebbian learning rules for single-layer associative memories to achieve similar capacities to Dense Associative Memories? Preliminary investigation into a subset of Hebbian learning rules found that the capacity characteristics broadly followed the derived capacity of traditional Hopfield networks (Lansner et al., 2025). But does this result hold true for all Hebbian learning rules?

3.1 Work Plan

Determining the capacity characteristics of Hebbian networks involves two components. The first is to provide theoretical reasoning as to the capacity of associative memories. The second is to provide experimental evidence of the theoretical reasoning on large-scale associative memory tasks: namely, pattern reconstruction and utility in a larger network.

3.1.1 Theoretical Work

We have seen above that we have proofs of the capacity of Hopfield networks and Dense Associative Memory. A contemporary example of this kind of proof can be found in (Bao et al., 2022), which proves capacity with respect to a perturbation of a query and a lower probability bound of error in recall.

Proving the capacity characteristics of Hebbian networks ought to follow the same formula. Given that the class of Hebbian update rules is large, we will focus in on a few learning rules: namely, the learning rules which were experimentally tested in Lansner et al. (2025), and then the expanded form in Definition 2.4.3. The goal of this part of the project is to come up with a theorem that shows that, with coefficients $c_1, c_2, \ldots, c_n$, that the Hebbian auto-associative memory has a capacity C which is some function of the dimensionality of the patterns stored (see Equations (1-6) in Lansner et al. (2025), or Theorem 1 from Bao et al. (2022)).

The proof strategy is the following: suppose that we have a learning rule with coefficients $c_1, c_2, \ldots, c_n$, denoted by $\mathcal{L}_{c_1,c_2,\ldots,c_n}$. Further, suppose that we have patterns $\xi^1, \xi^2, \ldots, \xi^N$, where each element of ξ^μ is either -1 or 1 with equal probability. Let $T(x)$ be the recall function of an associative memory with learning rule $\mathcal{L}_{c_1,c_2,\ldots,c_n}$. We define capacity in the same way as Bao et al. (2022), i.e., in relation to a perturbation constant $\delta < 0.5$ and an error threshold ε. We must prove the maximum N (number of patterns) which are all one-step (δ, ε)-recoverable. This means that N is the maximum number of patterns such that, for every pattern stored, the probability of a one-step recall of perturbation of the pattern not being equivalent to the pattern itself is less than ε:

$$\mathbb{P}(T(\bar{\xi}^i) \neq \xi^i) < \varepsilon, \ \forall i \in [1, N], \tag{20}$$

with $\bar{\xi}^i$ denoting the δ-perturbation of ξ^i.

Having a proof of this form is desirable as it allows to fairly compare Hebbian associative memories with other kinds of associative memories. Merely providing estimated scaling characteristics for the capacity of networks, as in Lansner et al. (2025), does not account for potential problematic elements in the kinds of patterns being memorized. For example, MNIST (LeCun et al., 2010) images, naively represented, are all highly correlated with one another. This "cross-talk" (Kohonen, 1988) can lead to spurious similarity

measures between query patterns and estimated patterns.

3.1.2 Experimental Work

Purely theoretical work is well and good. But the proof of the pudding is in the eating, and it is important to test the derived results for their accuracy to the real world. In order to verify our estimates, the goal will be to subject generated Hebbian associative memories to large-scale tasks. The most intuitive and popular task for associative memories is pattern reconstruction. Likewise, in order to see how these networks fair in real-world scenarios, integrating derived models in a larger architecture for the same task will provide an equal playing field for comparison.

Pattern reconstruction tasks are often included in the demonstration of associative memory capacities. Oftentimes, however, the demonstrations are merely qualitative. If the model can produce estimated patterns that look sufficiently similar to our eyes to the original query, then they succeed. But this is hardly objective, and so we require a better measure of reconstruction. In order to have an objective pattern reconstruction test, we need to have a shared basis in order to compare results. This is some distance metric for patterns, a function which gives us the similarity between a presented pattern and its output from the network. One common measure of similarity in Vector-Symbolic Architectures (Gayler, 1998; Plate, 1995; Smolensky, 1990) is *cosine similarity*:

$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\|\|y\|}, \tag{21}$$

which is the normalized dot-product between two vectors. This gives us the angle in D-dimensional space between the two vectors, where $\text{sim}(x, y) = 1$ means that the vectors are *orthogonal*. Preliminary experimentation with traditional Hopfield networks showed that, at or above critical capacity with the MNIST dataset, Hopfield models produce spurious states which have a high cosine similarity to all memorized patterns. To overcome this, it seems prudent to test similarity using a battery of similarity measures: for review, we can use those listed in Kelly et al. (2017).

4 Timeline

Theoretical work should be done by Christmas. Experimental work likewise requires only a couple of months. Data science certification requires emphasis here on large-scale projects. So, I would like to rigorously examine each model on a variety of tasks. One to two papers arises from this project: one talking about the theory of Hebbian DAM, and the other part including the experimental work. It would be better for a single paper, but if that runs too long, then experimental work can be a part 2.

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. (1987). Statistical mechanics of neural networks near saturation. *Annals of Physics*, *173*(1), 30–67. https://doi.org/10.1016/0003-4916(87)90092-3

Bao, H., Zhang, R., & Mao, Y. (2022). The capacity of the dense associative memory networks. *Neurocomputing*, *469*, 198–208. https://doi.org/10.1016/j.neucom.2021.10.058

Bi, G.-q., & Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of Neuroscience*, *18*(24), 10464–10472. https://doi.org/10.1523/JNEUROSCI.18-24-10464.1998

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

Gayler, R. W. (1998). *Multiplicative binding, representation operators & analogy (workshop poster)*. Retrieved October 10, 2025, from https://web-archive.southampton.ac.uk/cogprints.org/502/

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.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory* (1. issued in paperback). Routledge.

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

Hoover, B., Chau, D. H., Strobelt, H., Ram, P., & Krotov, D. (2024). Dense associative memory through the lens of random features [Version Number: 1]. https://doi.org/10.48550/ARXIV.2410.24153

Hoover, B., Liang, Y., Pham, B., Panda, R., Strobelt, H., Chau, D. H., Zaki, M. J., & Krotov, D. (2023, November 1). Energy transformer. https://doi.org/10.48550/arXiv.2302.07253

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, *81*(10), 3088–3092. https://doi.org/10.1073/pnas.81.10.3088

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., Blostein, D., & Mewhort, D. J. K. (2013). Encoding structure in holographic reduced representations [Place: US Publisher: Educational Publishing Foundation]. *Canadian Journal of Experimental Psychology / Revue canadienne de psychologie expérimentale*, *67*(2), 79–93. https://doi.org/10.1037/a0030301

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

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

Lansner, A., Ravichandran, N. B., & Herman, P. (2025, February 17). Benchmarking hebbian learning rules for associative memory. https://doi.org/10.48550/arXiv.2401.00335

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

Markram, H., Lübke, J., Frotscher, M., & Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, *275*(5297), 213–215. https://doi.org/10.1126/science.275.5297.213

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

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. (2006). How far can you go with hebbian learning, and when does it lead you astray? In Y. Munakata & M. H. Johnson (Eds.), *Processes of change in brain and cognitive development* (pp. 33–60). Oxford University PressOxford. https://doi.org/10.1093/oso/9780198568742.003.0002

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., Salvatori, T., Song, Y., Lukasiewicz, T., & Bogacz, R. (2022, June 17). Universal hopfield networks: A general framework for single-shot associative memory models. https://doi.org/10.48550/arXiv.2202.04557

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

Plate, T. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6*(3), 623–641. https://doi.org/10.1109/72.377968

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

Rolls, E. T. (2013). The mechanisms for pattern completion and pattern separation in the hippocampus. *Frontiers in Systems Neuroscience*, *7*. https://doi.org/10.3389/fnsys.2013.00074

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

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Salvatori, T., Song, Y., Hong, Y., Frieder, S., Sha, L., Xu, Z., Bogacz, R., & Lukasiewicz, T. (2021, September 16). Associative memories via predictive coding. https://doi.org/10.48550/arXiv.2109.08063

Sejnowski, T. J., & Tesauro, G. (1989). The hebb rule for synaptic plasticity: Algorithms and. In *Neural models of plasticity*. Academic Press.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*(1), 159–216. https://doi.org/10.1016/0004-3702(90)90007-M

Spratling, M. W. (2017). A review of predictive coding algorithms. *Brain and Cognition*, *112*, 92–97. https://doi.org/10.1016/j.bandc.2015.11.003

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

Wu, D., Hu, J. Y.-C., Hsiao, T.-Y., & Liu, H. (2024, November 10). Uniform memory retrieval with larger capacity for modern hopfield models. https://doi.org/10.48550/arXiv.2404.03827