# Energy-based Associative Memories

Connor Hanley

September 22, 2025

# Outline of Presentation

# Associative Memories

## Definition (Associative memory)

An *associative memory* is a 3-tuple $\langle f, A, C \rangle$ obeying the following properties[a]:

(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 $y$.

[a] Following the definition from (Kanerva, 1993)

(a) Associative memories learn to associate patterns $(x, y)$ in the learning phase

(b) Typically the task is to recall $y$ based on noisy, masked, or degraded forms of $x$

(c) An associative memory is said to be *auto-associative* if $x = y$ (therefore $A = C$)

(d) It is said to be *hetero-associative* otherwise

# Correlation Matrix based AMs

(a) There are many different kinds of associative memories

(b) The kind of memory that we will be focusing on here are *correlation matrix* based memories (Amari, 1972; J. J. Hopfield, 1984; J. Hopfield, 1982; Kohonen, 1988; Nakano, 1972)

## Definition (Correlation Matrix AM)

A *correlation matrix* AM is a tuple $\langle f, A, C \rangle$ with:

(i) Standard $N \times D$ address matrix,

(ii) Standard $N \times M$ content matrix,

(iii) Recall function:

$$f_{A,C}(x) = g(Wx),$$
$$\text{with } W = C^\top A.$$

The function $g$ in (iii) is either:

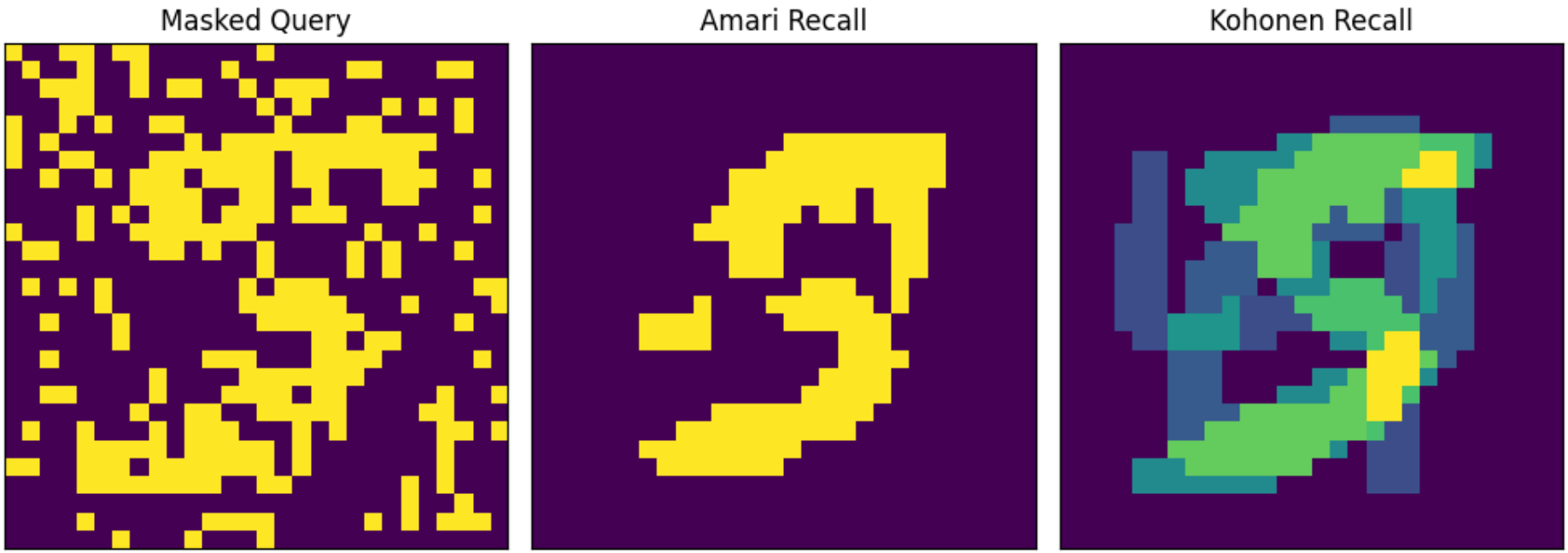
(a) In Kohonen (1988) the identity function $g(x) = x$

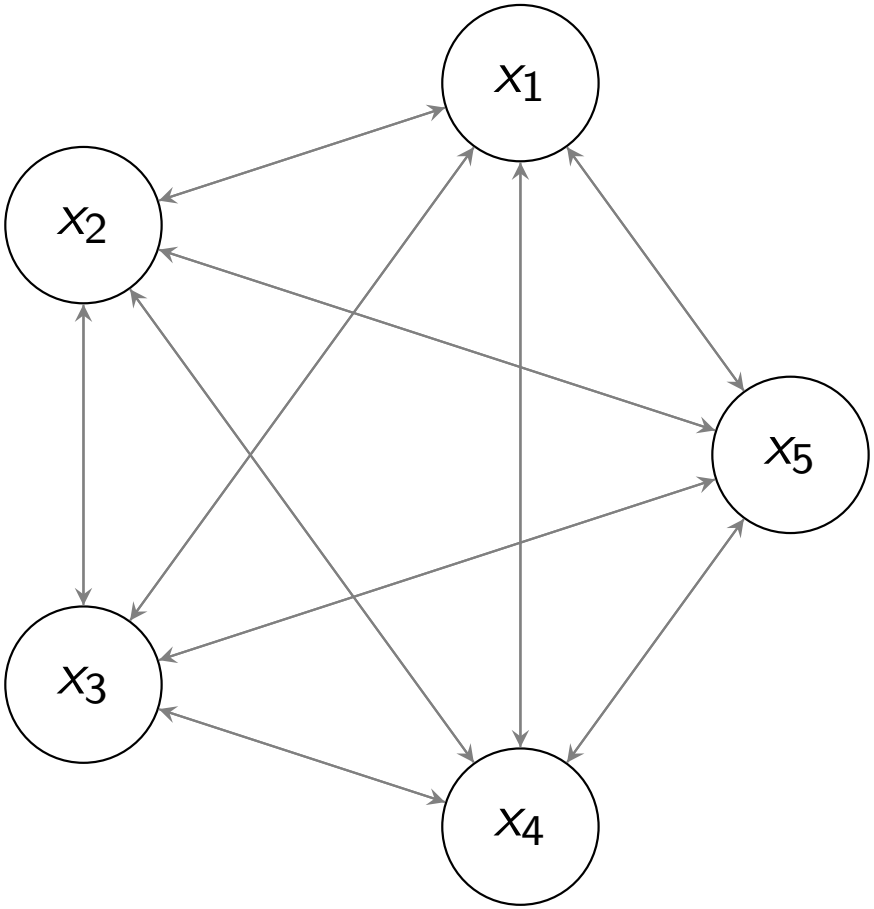
(b) In Amari (1972), J. Hopfield (1982), and Nakano (1972), $g(x) = \operatorname{sgn}\{x\}$, with

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

called the *signum* function.

# Kohonen Network versus Amari Network






Where each connection is $w_{ij} = w_{ji}$.

# Hebbian Update Rules

## Definition (Hebbian update-rule; Hebbian learning)

Let $x_n$, $n = 1, 2, \ldots, L$ be a *layer* of computational units, up to $L$ layers. Let the weights $W_{ij}$ be a matrix representing the weights of full connections between layers $x_i$ and $x_j$. Then, we say that a weight update rule is *Hebbian* iff

$$\Delta W_{ij} = f(x_i, x_j).$$

## Example (Activity Product Rule)

The simplest form of Hebbian learning is the following update rule[a]:

$$\Delta W_{ij} = \eta x_j x_i^\top,$$

where $\eta$ is the *learning rate*.

[a] (Haykin, 2009)

# Correlation Matrix Memories are Hebbian

We can formulate Correlation Matrix AM learning of patterns as Hebbian. Let $\xi_\mu$, $\mu = 1, 2, \ldots, N$ be the $N$ patterns of $D$ dimension that we want the system to memorize. For each time step, suppose we present a new pattern $\xi_{t=\mu}$:

$$W^{(0)} \leftarrow [0]_{D\times D},$$
$$W^{(t+1)} \leftarrow W^{(t)} + \eta \xi_i \xi_i^\top.$$

Formulated as a difference,

$$\Delta W = \eta \xi_i \xi_i^\top.$$

At time $t = \mu$, this is equivalent to the matrix:

$$W = \sum_{\mu=1}^{N} \xi_\mu \xi_\mu^\top = \Xi^\top \Xi,$$

with $N \times D$ pattern matrix $\Xi = [\xi_1, \xi_2, \ldots, \xi_N]$.

# Energy and Correlation Matrix (Hopfield!)

Correlation Matrix AMs can be understood as performing *energy minimization*. In the same way that we can specify the *weight dynamics* with the update rule, we can also specify the *neural dynamics* of the Hopfield network; i.e., asynchronous update rules.

## Definition (Hopfield energy)

Let $\Xi = [\xi_1, \xi_2, \ldots, \xi_N]$ be the $N \times D$ pattern matrix. Let $W = \Xi^T \Xi$. Let $\sigma$ be a $D$-dimensional query vector. The *energy* (J. J. Hopfield, 1984; Krotov & Hopfield, 2016) of the AM given the query vector $\sigma$ is the *dot product correlations* between the weights and the query vector:

$$E_{\text{Hopfield}} = -\frac{1}{2}\sum_{i,j}^{N} W_{ij}\sigma_i\sigma_j = -\frac{1}{2}\sum_{\mu=1}^{N}\left(\sum_{i=1}^{D}\Xi_{\mu i}\sigma_i\right)^2 .$$

## Energy intuition

The **energy** of a Hopfield network is the *weighted* dot product between each computational unit and every other. Low energy means there is *high* correlation between the value of each comptuational unit and the others. High energy means that there is *low* correlation between each computational unit and every other.

# Asynchronous update rule

In the energy-based framework, the AM is updated *asynchronously*. At each time step $t$, we pick a random neuron and flip its value. If this value *lowers* the energy, then we keep the flip. Otherwise, we pick another random value.

## Definition (Asynchronous Hopfield update rule)

$$\sigma_i^{(t+1)} = \text{argmin}_{b \in \{-1,1\}} [E_{\text{Hopfield}}(\sigma_i')]$$

with $\sigma' = \sigma^{(t)}$, but with the $i$'th element set to $b$.

## Fact (One-step rule relation to asynchronous rule)

*The asynchronous rule defined above relates to the single-pass update rule as you can derive the single-pass rule from the difference of the energies of the query buffer with the i'th unit flipped and unflipped. (Krotov et al., 2025; Mimura et al., 2025).*

# Energy Dynamics

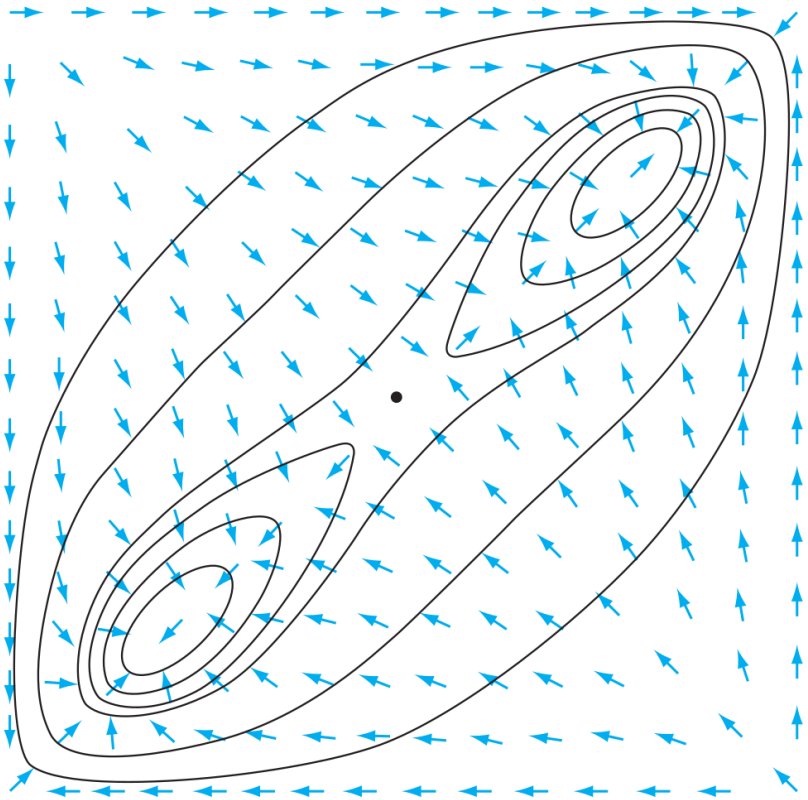
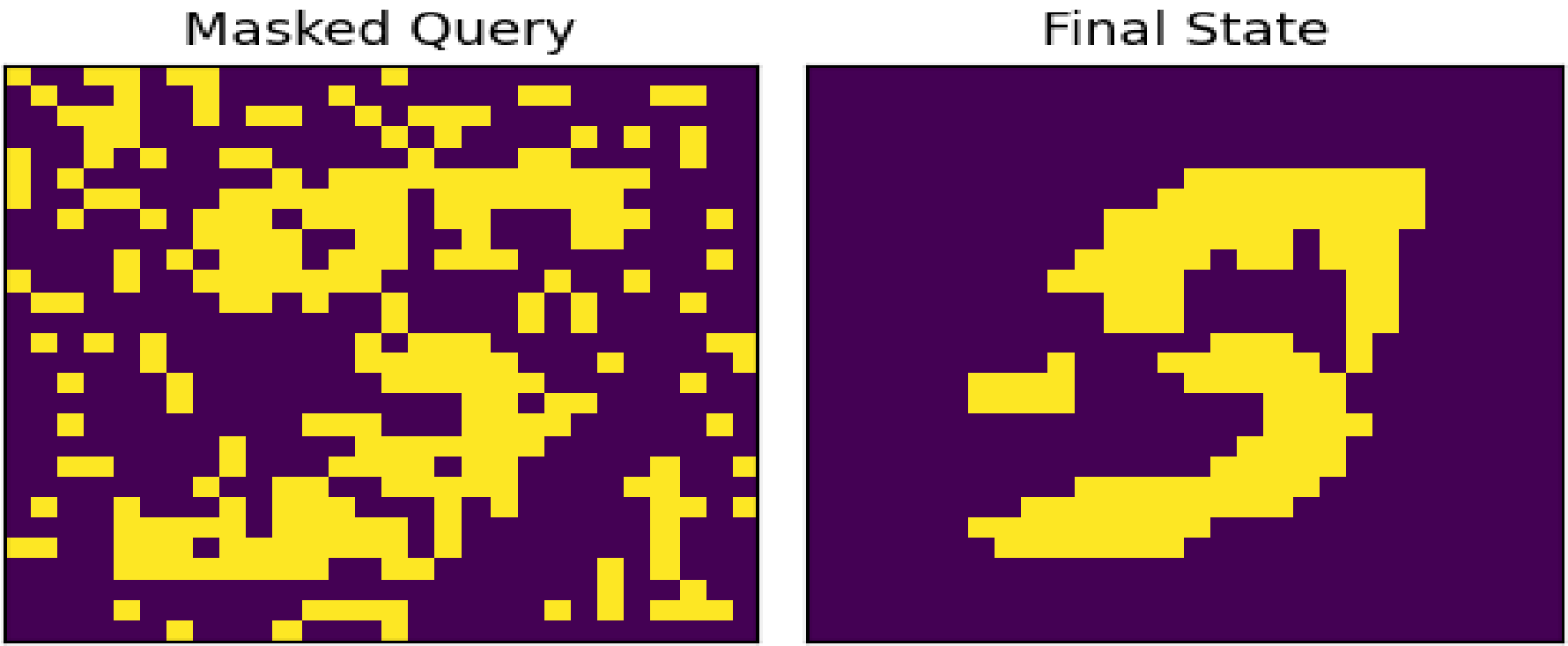
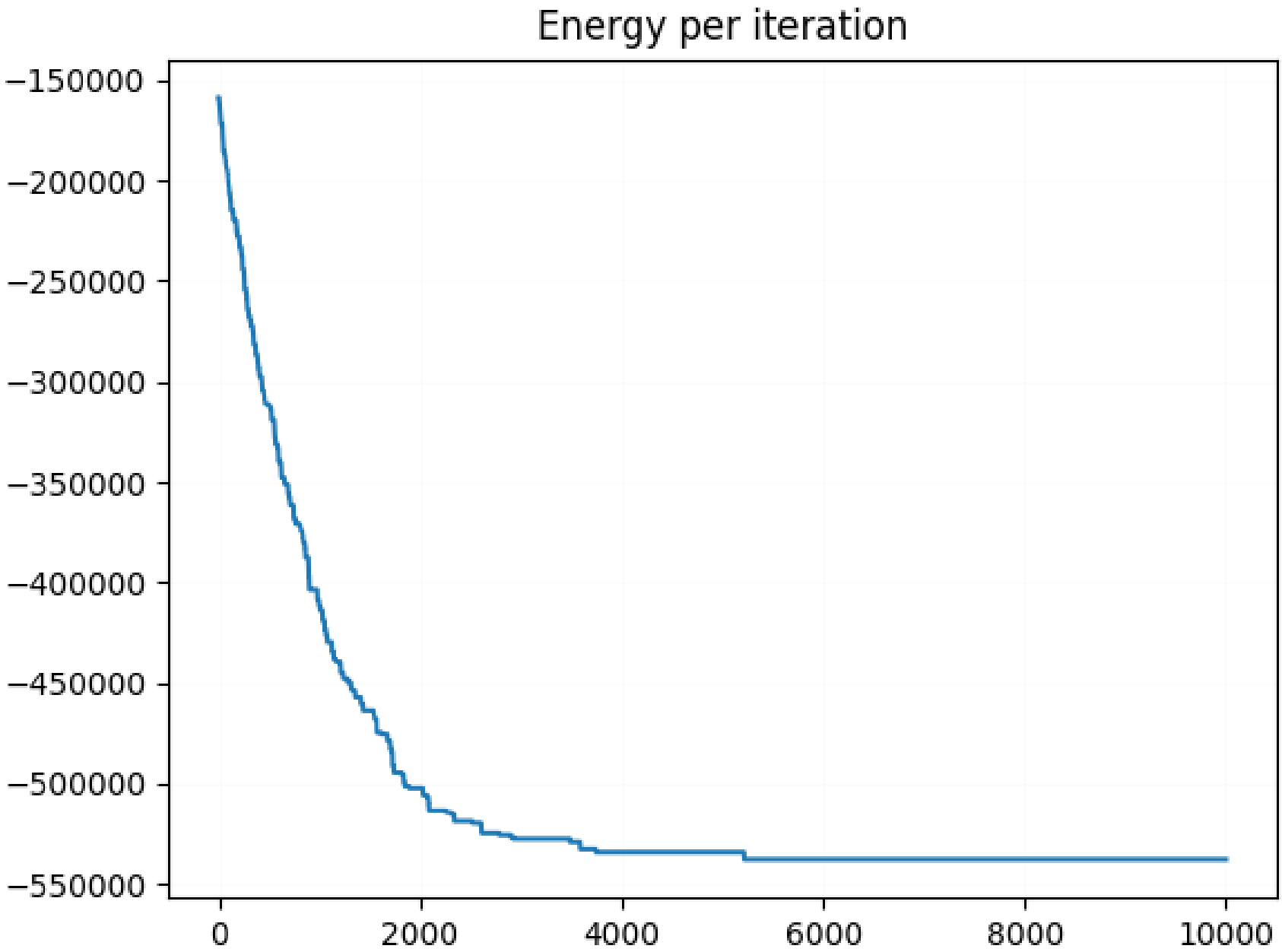


Figure: "Neurodynamics" of the Hopfield energy function. From Haykin (2009), originally from J. J. Hopfield (1984).

# Energy-based Recall (Demonstration)




Energy per iteration
−150000
−200000
−250000
−300000
−350000
−400000
−450000
−500000
−550000
0
2000
4000
6000
8000
10000

# Increasing storage capacity with Dense Associative Memories

Convergent investigation into Dense Associative Memory-like models began soon after J. Hopfield (1982) with Hintzman (1984). Tensor-based energy functions and weights with Chen et al. (1986) and Psaltis and Cheol Hoon Park (1986) (see also, Kelly et al. (2017)).

Hopfield networks famously have a capacity for $\sim 0.14D$ memories, where $D$ is the dimension of the patterns to store; note that this is can be even further diminished by correlated patterns, introducing "cross-talk" (note even in Kohonen (1988))

# Generalized Energy Function

## Definition (Generalized Energy Function)

Consider the following abstraction of $E_{\text{Hopfield}}$:

$$E_{F_n} = -\sum_{\mu=1}^{N} F\left(\sum_{i=1}^{D} \xi_{\mu i}\sigma_i\right)$$

We call $F_n$ the *Lagrangian* function (Krotov, 2021). $F_n$ is of the form $\frac{1}{n}(x)^n$.

We can also get an update rule from this:

## Definition (Generalized Update Function)

For energy function with Lagrangian $F_n$, $E_{F_n}$, the single-pass update function is (Demircigil et al., 2017; Krotov, 2021; Krotov & Hopfield, 2016):

$$\sigma^{(t+1)} = \text{sgn}\left\{\sum_{\mu=1}^{N} \xi_{\mu i} F_n'\left(\sum_{j\neq i}^{D} \xi_{\mu j}\sigma_j\right)\right\},$$

equivalently,

$$\sigma^{(t+1)} = \text{sgn}\left\{\Xi^T F'\left(\Xi\sigma\right)\right\}$$

## Example (Classical Hopfield)

The above definition gets us the classical Hopfield energy function immediately, with energy,

$$E_{\text{Hopfield}} = E_{F_2} = -\frac{1}{2}\sum_{\mu=1}^{N}\left(\sum_{i=1}^{D}\xi_{\mu i}\sigma_i\right)^2;$$

and update:

$$\sigma^{(t+1)} = \text{sgn}\left\{\Xi^T\Xi\sigma^{(t)}\right\}.$$

## Example (Minerva2)

We can also get Minerva2 (and more generally, every Minerva model) (Hintzman, 1984; Kelly et al., 2017): [a]

$$E_{\text{Minerva2}} = E_{F_4} = -\frac{1}{4}\sum_{\mu=1}^{N}\left(\sum_{i=1}^{D}\xi_{\mu i}\sigma_i\right)^4;$$

and update rule:

$$\sigma^{(t+1)} = \text{sgn}\left\{\Xi^T\left(\Xi\sigma^{(t)}\right)^3\right\}.$$

[a] To see further corresp. with Memory Tesseract, note that $F_4$ gives 4-way interaction term in the weights.

# Capacity

There are many ways to characterize the capacity of an AM; note that since it is totally defined, we won't get "out-of-bounds" memory access or null-pointers

The trade off is we get a "crap-in-crap-out" phenomenon: if the memory is filled with too many items, we enter into *spurious* energy states

Dense Associative Memory theory gives us the following relation:

**Theorem (Capacity scaling laws)**

*We say that $N^{max}$ is the number of patterns the AM can store ( literature uses $K$, see Krotov and Hopfield (2016) ). With Lagrangian function $F_n$, $N^{max}$ scales exponentially with $n$:*

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

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

## Lemma (Capacity of Minerva2)

*We immediately get from the above that the capacity of Minerva2 is:*

$$N^{max} \propto N^3.$$

# Modern Hopfield Networks

So far we have been dealing with bipolar values $x \in \{-1, 1\}$: but this perspective can be expanded to real values. Namely, Modern Hopfield Networks (Ramsauer et al., 2021) which are equivalent to Multi-head Attention in Transformers (Vaswani et al., 2023).

## Definition (Modern Hopfield)

Energy of multi-head attention relies on the Lagrangian function *logsumexp*, defined as:

$$\mathsf{lse}(\beta, x) = \frac{1}{\beta} \log \left( \sum_{i=1}^{D} \exp(\beta x_i) \right).$$

This gives us an update rule

$$\sigma^{(t+1)} = \Xi^T \mathsf{softmax} \left( \beta \Xi \sigma^{(t)} \right).$$

# If you're interested in this

(a) This was just the summary of general results

(b) There is a new framework for architecture agnostic energy-descent networks, called HAMUX (Krotov, 2021)

(c) HAMUX gets you compositional energy functions, and an energy-descent rule which is in terms of all of the different parts of the network

(d) I'm not too familiar with it, so I didn't talk about it here.

# DAM Not Hebbian

(a) Closing remarks: DAM is not Hebbian. This is because the weight update rule requires multiple interaction terms, again see Kelly et al. (2017).

(b) Training usually proceeds using conventional methods: backpropagation or even predictive coding (Millidge et al., 2022; Salvatori et al., 2021)

(c) can we make a Hebbian network which has similar scaling laws and capacity as DAMs?

# Bibliography I


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

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

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

# Bibliography II


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. 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.

Kanerva, P. (1993). Sparse distributed memory and related models. In *Associative neural memories: Theory and implementation* (pp. 50–76). Oxford University Press.

# Bibliography III


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, April 7). 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. J. (2016, September 27). Dense associative memory for pattern recognition. https://doi.org/10.48550/arXiv.1606.01164

# Bibliography IV


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

Mimura, K., Takeuchi, J., Sumikawa, Y., Kabashima, Y., & Coolen, A. C. C. (2025, June 1). Dynamical properties of dense associative memory. https://doi.org/10.48550/arXiv.2506.00851

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

Psaltis, D., & Cheol Hoon Park. (1986). Nonlinear discriminant functions and associative memories [ISSN: 0094243X]. *AIP Conference Proceedings*, *151*, 370–375. https://doi.org/10.1063/1.36241

# Bibliography V


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

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

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