Hopfield Networks as a model of auto-associative memories

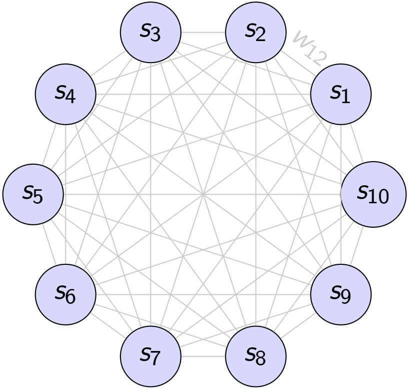
Masters in Brain & Cognition, UPF

Alex Hyafil

Context and Motivation

- Introduced by **John Hopfield** (1982)
- One of the first neural networks with a theoretical foundation
- Strong links to:
  - Statistical physics (Ising models)
  - Dynamical systems
  - Energy minimization
- Contributed to the **2024 Nobel Prize in Physics**

Network Architecture



- Binary neuron states $s_i \in \{-1, +1\}$
- Fully connected recurrent architecture
- Symmetric weights $w_{ij} = w_{ji}$

Network connectivity stores learned patterns

The network stores patterns P patterns:

$$\xi^\mu = (\xi_1^\mu, \dots, \xi_N^\mu)$$

Weights are:

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu, \quad i \neq j$$

- The weight between neuron i and j encodes the correlation between ξ_i and ξ_j across patterns (you might say that the connectivity matrix encodes the structure of the world)
- Corresponds to **Hebbian learning rule** applied after clamping sequentially each pattern sequentially

Neuron Update Rule

- Local field:

$$h_i = \sum_j w_{ij} s_j$$

- Update rule for neuron i:

$$s_i \leftarrow \mathrm{sign}(h_i)$$

- Neurons can be updated either synchronously (all at same time) or asynchronously (one at a time)

Energy Function and Monotonic Decrease

Energy of a network state:

$$E(\vec{s}) = -\frac{1}{2} \sum_{i \neq j} w_{ij} s_i s_j$$

Each asynchronous update never increases the energy

<u>Proof:</u> Consider updating neuron i asynchronously:

$$s_i' = \mathrm{sign}(h_i), \quad h_i = \sum_j w_{ij} s_j$$

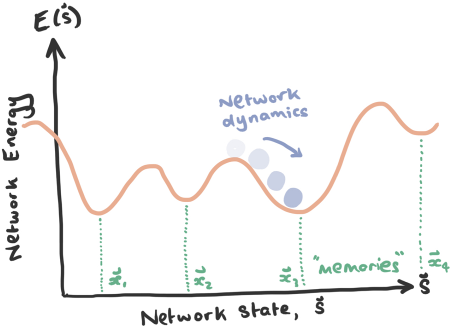
Only terms involving s_i change:

$$\Delta E = E(s_i') - E(s_i) = -(s_i' - s_i) h_i$$

Since $h_i = |h_i| \, \mathrm{sign}(h_i) = |h_i| s_i'$, we obtain:

$$\Delta E = -|h_i| \left(1 - s_i s_i'\right) \leq 0$$

Energy Landscape: Attractors



- Local minima correspond to stable fixed points - under certain conditions correspond to stored patterns
- Valleys define basins of attraction

Overlap with Stored Patterns

Define the overlap as a measure of similarity between current state $\vec{s}$ and pattern ξ^{μ}:

$$m^{\mu} = \frac{1}{N}\sum_{i=1}^{N} \xi_i^{\mu} s_i, -1 \leq m^{\mu} \leq 1$$

where $m^{\mu} = 1$ means full overlap with pattern ξ^{μ} and $m^{\mu} = -1$ means full overlap with anti-pattern (-1/+1 flipped)
Activity can be rewritten as $h_i = \sum_j w_{ij} s_j = \frac{1}{N}\sum_{j,\mu} \xi_i^{\mu}\xi_j^{\mu} s_j = \sum_{\mu} \xi_i^{\mu} m^{\mu}$

If patterns ξ^{μ} are random and uncorrelated:
Starting at $\vec{s}(t) = \xi^{\mu}$, $m^{\mu}(t) = 1$ and $m^{\mu}(t) \propto < \xi^{\mu}\xi^{\nu} > = 0$ for $\mu \neq \nu$

$$h_i = \xi_i^{\mu} m^{\mu} + \sum_{\mu \neq \mu} \xi_i^{\nu} m^{\nu} = \xi_i^{\mu} = s_i(t)$$

so $s_i(t+1) = \text{sign}(h_i) = s_i(t)$

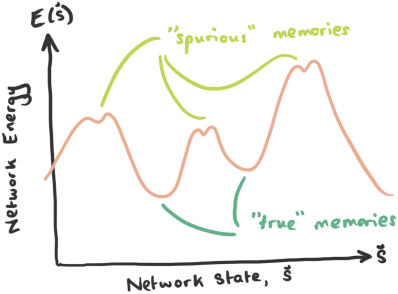
Stored patterns are fixed points of the overlap dynamics

Capacity and Limitations

- A network can only store a finite amount of patterns
- Maximum random patterns stored scales **linearly with network size** N:

$$P_{\text{max}} \approx 0.138\, N$$

- Too many memories reduce the size of the basin of attractions
- Spurious attractors appear (e.g. mixtures of patterns)



Boltzmann Machines

- Ppopularized by Geoff Hinton (received Nobel Prize along with Hopfield)
- Neurons update probabilistically:

$$P(s_i = +1) = \sigma\left(\frac{1}{T}\sum_j w_{ij} s_j\right)$$

- If temperature T goes to 0, we're back to a Hopfield network.
- Equivalent to sampling from:

$$p(\vec{s}) = \frac{1}{Z} e^{-E(\vec{s})/T}$$

- Basis for probabilistic generative models

Modern Hopfield Networks

- Recent reformulation of Hopfield networks using continuous states
- Energy function with **exponentially large capacity**
- Retrieval corresponds to softmax-based attention

Key idea: replace binary neurons by continuous states and use an energy of the form

$$E(\vec{s}) = -\log \sum_{\mu} \exp(\beta\, \xi^{\mu} \cdot \vec{s})$$

- Fixed points correspond to stored patterns
- Closely related to transformer attention mechanisms
- Bridge between associative memory and modern deep learning

Further reading:

- `https://ml-jku.github.io/hopfield-layers/`

Further reading:

- https://neuronaldynamics.epfl.ch/online/Ch17.S2.html
- https://ml-jku.github.io/hopfield-layers/