Hopfield Networks

Nan Ye

School of Mathematics and Physics
The University of Queensland

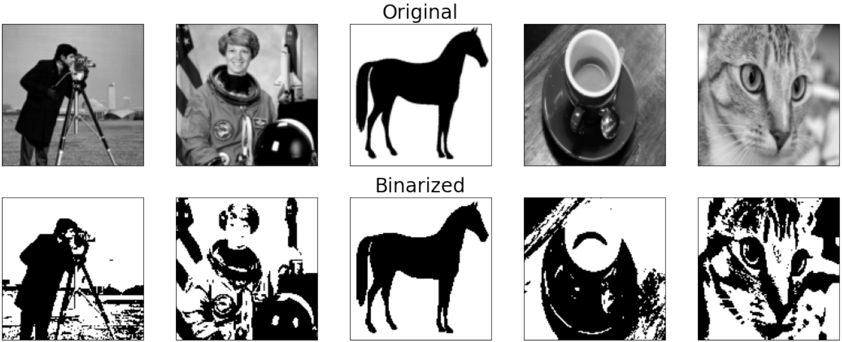
Hopfield Nets

- Developed by William Little (1974) and John Hopfield (1982).
- Hopfield nets are neural nets designed as a model of human memory.
  - Architecture: a collection of linear threshold units (Perceptrons) connected to each other.
  - Memorization: adjust the network weights to ‘remember’ input patterns.
  - Retrieval: initialise neuron states to be a partial or noisy input, then use the network weights to perform computation to retrieve the matching pattern in memory.
- In other terms, Hopfield nets serve as content-addressable or associative memory systems with binary threshold nodes.

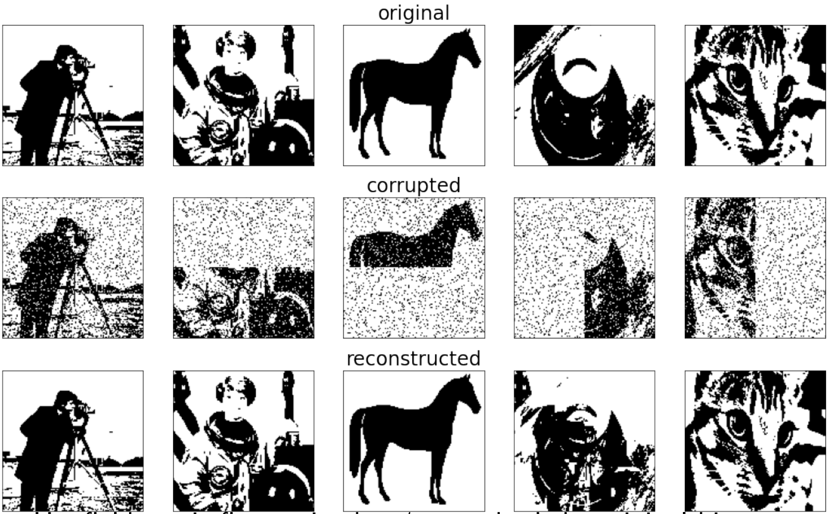
Little, The existence of persistent states in the brain, 1974

Hopfield, Neural networks and physical systems with emergent collective computational abilities, 1982

Illustration



- Top row: some images that we pick.
- Bottom row: binarized versions of the chosen images
  - We use binary images because a Hopfield net can only ‘remember’ binary images.



- Hopfield net is first trained to 'remember' the original binary images, then it can 'recall' the original image given a partial/noisy version of it.

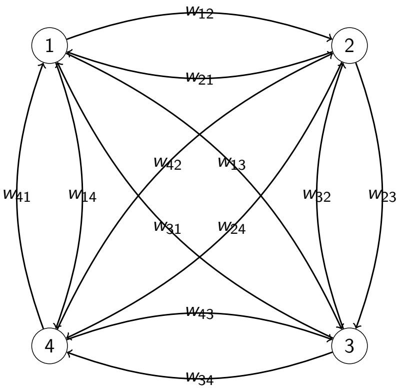
How Does a Hopfield Net Work

- The activation state of each neuron in the network represents one bit of the current pattern that the network is ‘thinking’ about.
- Given a set of patterns, the weights of the network are first trained to “memorize” the pattern.
- Given a partial or noisy pattern, the network initialises its neuron activation states to the given pattern, and then run updates until convergence.

Network Structure

- We want to deal with binary patterns consisting of m -1 or $+1$.
- We need m neurons in the Hopfield network.
- Each is a linear threshold unit, with all other neurons' outputs (activation states) as the inputs.
  - The weight for neuron i as an input to neuron j is w_{ij}.
  - The weights are symmetric, that is, $w_{ij} = w_{ji}$.
  - The output of neuron i is 1 if the weighted sum is ≥ 0 and -1 otherwise.

- A Hopfield net is a recurrent network, i.e., there are cycles in the architectural graph.



A Hopfield net with 4 neurons

Training

Hebbian learning (repetition reinforces a synapse)

- Initially, set all weights to 0.
- Given a pattern $\mathbf{a} = (a_1, \ldots, a_m) \in \{-1, +1\}^m$, the network updates each weight w_{ij} for $i \neq j$ using

  $$w_{ij} \leftarrow w_{ij} + a_i a_j$$

  Equivalently, update perceptron i with example $(\mathbf{a}, a_i)$.
- Remarks
  - The connection between i and j is strengthened if both units are on, and is weakened otherwise.
  - The weights remain symmetric.
  - This allows learning to remember patterns in an incremental way.

Hopfield's weight formula

- Assume we have n patterns $\mathbf{a}_1, \ldots, \mathbf{a}_n$ with $\mathbf{a}_i = (a_{i1}, \ldots, a_{im})$, then the weights are set as follows

$$w_{ij} = \begin{cases} \sum_{s=1}^{n} a_{si} a_{sj}, & i \neq j, \\ 0, & i = j. \end{cases}$$

Example: one pattern only

- Assume we have only one pattern $(-1, 1, 1, -1, 1)$.
- Then $w_{12} = -1 \cdot 1 = -1$, $w_{13} = -1 \cdot 1$ and so on.
- The complete weight matrix is given by

$$(w_{ij}) = \begin{pmatrix} 0 & -1 & -1 & 1 & -1 \\ -1 & 0 & 1 & -1 & 1 \\ -1 & 1 & 0 & -1 & 1 \\ 1 & -1 & -1 & 0 & -1 \\ -1 & 1 & 1 & -1 & 0 \end{pmatrix}$$

Example: two patterns

- Assume we have two patterns $(-1, 1, 1, -1, 1)$ and $(1, -1, 1, -1, 1)$, then the weight matrix is

$$(w_{ij}) = \begin{pmatrix} 0 & -2 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 2 \\ 0 & 0 & -2 & 0 & -2 \\ 0 & 0 & 2 & -2 & 0 \end{pmatrix}$$

Retrieving a Pattern

- Given a partial or noisy pattern $\mathbf{a} = (a_1, \ldots, a_m)$.
- We first set the activation state of each neuron to the corresponding a_i.
- Now repeatedly update the activation states of the neurons until they don't change
  - we need to decide the order of the updates — there are different ways to do this (discussed later)
  - when the i-th neuron is chosen to be updated, we simply recompute its activation

$$a_i \leftarrow \mathrm{sgn}(w_{\cdot i} \cdot \mathbf{a})$$

Example

- Consider the previous network trained with two patterns.
- Assume we are given a pattern $(1, 1, 1, 1, 1)$.
- The updated value of a_3 is

$$\begin{aligned}
\mathrm{sgn}(w_{\cdot 3} \cdot \mathbf{a}) &= \mathrm{sgn}(w_{13}a_1 + w_{23}a_2 + w_{33}a_3 + w_{43}a_4 + w_{53}a_5) \\
&= \mathrm{sgn}(0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + (-2) \cdot 1 + 2 \cdot 1) \\
&= 1
\end{aligned}$$

Sequencing the Updates

- Synchronous update: all nodes are updated at the same time
  - biologically not realistic as neurons may update at different rates.
- Asynchronous update: randomly select a neuron and then update it
- Semi-random update: update all nodes in one step, but update nodes in a random order
  - commonly used in practice

Finishing off the example

- We have the network trained with two patterns.
- We are given the pattern $(1, 1, 1, 1, 1)$.
- Update the nodes in the following order

$$3, 1, 5, 2, 4, 3, 1, 5, 2, 4, \ldots$$

- What's the final pattern?

Update	New pattern
$a'_3 = \mathrm{sgn}((0,0,0,-2,2)\cdot(1,1,1,1,1)) = \mathrm{sgn}(0) = 1$	$(1,1,1,1,1)$
$a'_1 = \mathrm{sgn}((0,-2,0,0,0)\cdot(1,1,1,1,1)) = \mathrm{sgn}(-2) = -1$	$(-1,1,1,1,1)$
$a'_5 = \mathrm{sgn}((0,0,2,-2,0)\cdot(-1,1,1,1,1)) = \mathrm{sgn}(0) = 1$	$(-1,1,1,1,1)$
$a'_2 = \mathrm{sgn}((-2,0,0,0,0)\cdot(-1,1,1,1,1)) = \mathrm{sgn}(2) = 1$	$(-1,1,1,1,1)$
$a'_4 = \mathrm{sgn}((0,0,-2,0,-2)\cdot(-1,1,1,1,1)) = \mathrm{sgn}(-4) = -1$	$(-1,1,1,-1,1)$

Doing this one more iteration shows that the pattern does not change. So we recover the pattern (-1, 1, 1, -1, 1).

Memory and Energy (Optional)

- The energy of the current activation state $\mathbf{a}$ of a Hopfield net is

$$E_{\mathbf{w}}(\mathbf{a}) = -\frac{1}{2}\sum_{i,j} w_{ij} a_i a_j,$$

  where $\mathbf{w}$ is the weight matrix (w_{ij}).
- Memorization: choose $\mathbf{w}$ so that each interested pattern $\mathbf{a}$ is likely to be a local minimizer of $E_{\mathbf{w}}$.
  - Local minimizer: a.k.a. attractor, stable pattern
- Retrieval: given a partial/noisy pattern $\mathbf{a}'$, move it towards a local minimizer of $E_{\mathbf{w}}$.

Your Turn

Which of the following statement is correct? (Multiple choice)

(a) A Hopfield net is proposed as a model of human memory.

(b) A Hopfield net is a recurrent neural net.

(c) The weight matrix of a Hopfield net is symmetric with 0's on the diagonal.

What You Need to Know

- A Hopfield net is inspired by how human memory works.
- A Hopfield net has a recurrent architecture.
- Memorization using Hebbian learning or Hopfield's formula.
- Retrieval using synchronous, asynchronous, or semi-random updates.