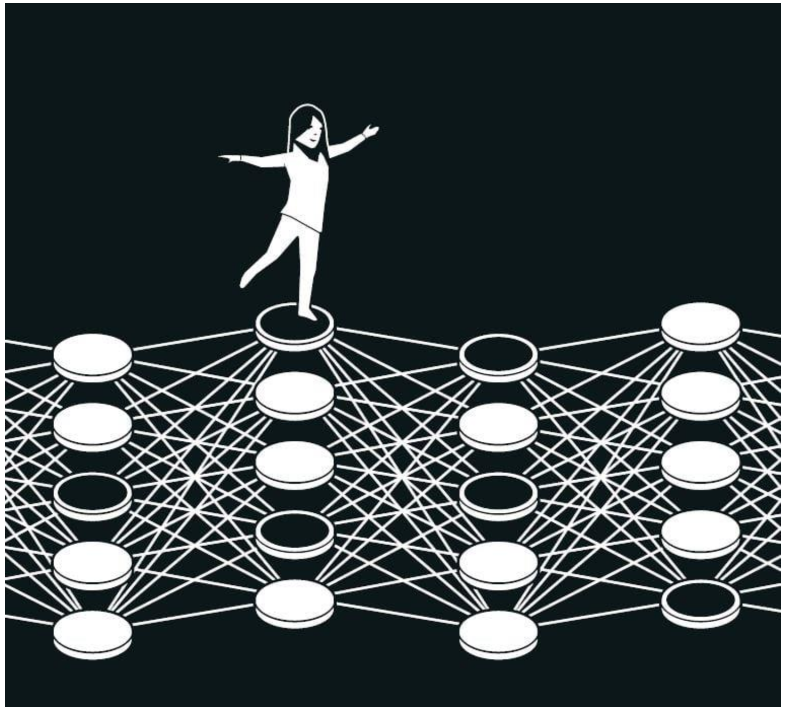
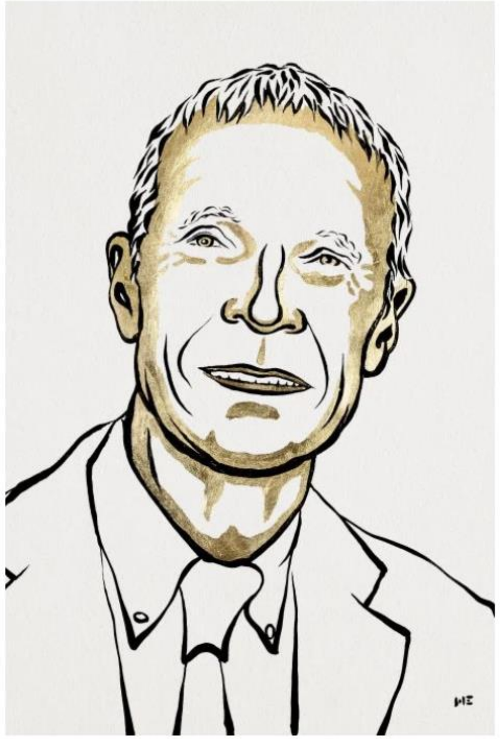
Amari-Hopfield networks



2024/12/18

Tatsuo Okubo

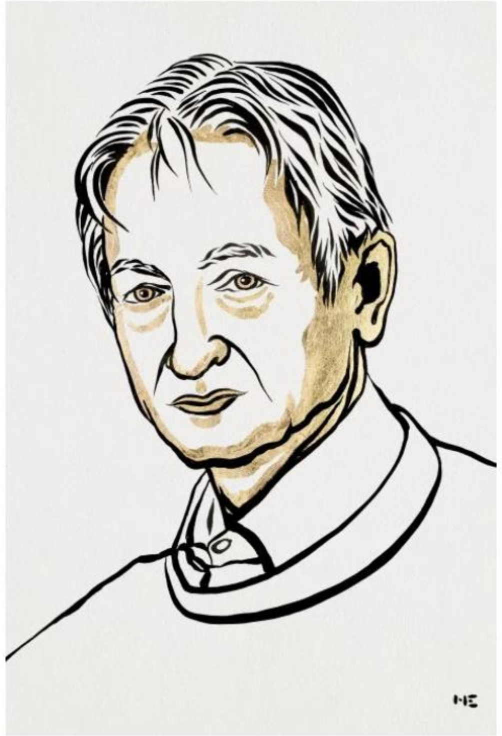
The Nobel Prize in Physics 2024



Ill. Niklas Elmehed © Nobel Prize Outreach

John J. Hopfield

Prize share: 1/2



Ill. Niklas Elmehed © Nobel Prize Outreach

Geoffrey Hinton

Prize share: 1/2

“for foundational discoveries and inventions that enable machine learning with artificial neural networks”

Proc. Natl. Acad. Sci. USA
Vol. 79, pp. 2554–2558, April 1982
Biophysics

Neural networks and physical systems with emergent collective computational abilities

(associative memory/parallel processing/categorization/content-addressable memory/fail-soft devices)

J. J. HOPFIELD

Division of Chemistry and Biology, California Institute of Technology, Pasadena, California 91125; and Bell Laboratories, Murray Hill, New Jersey 07974

Contributed by John J. Hopfield, January 15, 1982

Neural networks and physical systems with emergent collective computational abilities.

JJ Hopfield - Proceedings of the national academy of …, 1982 - National Acad Sciences

Computational properties of use of biological organisms or to the construction of computers can emerge as collective properties of systems having a large number of simple equivalent …

Amari's model (1972)

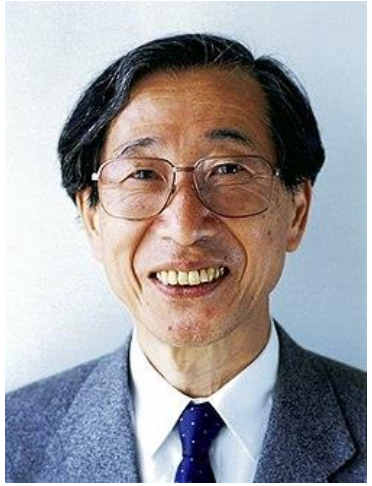
IEEE TRANSACTIONS ON COMPUTERS, VOL. C-21, NO. 11, NOVEMBER 1972 1197

Learning Patterns and Pattern Sequences by Self-Organizing Nets of Threshold Elements

SHUN-ICHI AMARI

$$x_i' = \operatorname{sgn}\left(\sum_j w_{ij}x_j - h_i\right)$$

$$W = \sum_\alpha \tilde{\lambda}_\alpha \mathbf{x}_\alpha \mathbf{x}_\alpha{}^t + \sigma^2 E.$$



甘利俊一

Amari's works are cited in Hopfield (1982),
but somehow not the correct one.

6. Amari, S.-I. (1977) *Biol. Cybern.* **26**, 175–185.
7. Amari, S.-I. & Akikazu, T. (1978) *Biol. Cybern.* **29**, 127–136.

Components of a Amari-Hopfield model

- Function of the network
- Single neuron dynamics
- Memory formation
- Energy
- Memory capacity

1. Function

What can a Hopfield network do?

Function #1: memory

- Content-addressable memory
  - memory that allows data to be accessed based on its content rather than its storage location

- pattern completion

figure from Kleinfeld (Neuron, 2024)

Function #2: solve optimization problems

- Energy minimization -> minimizing objective function

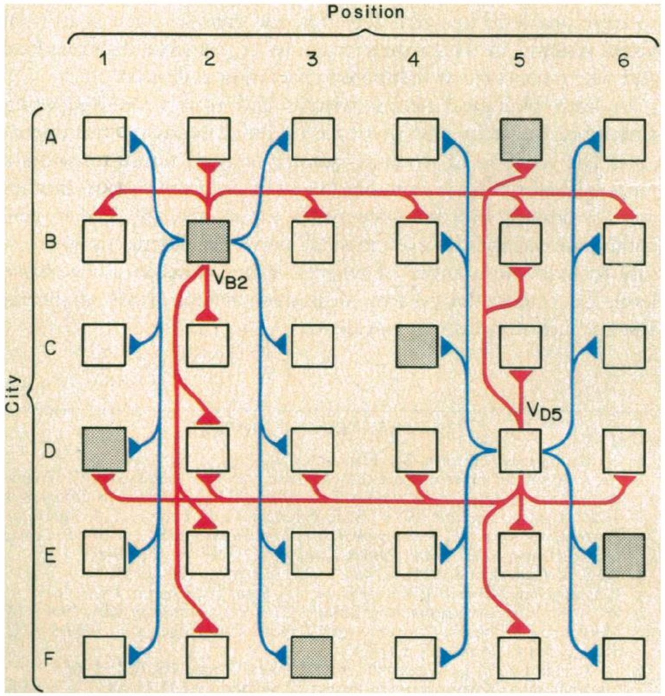
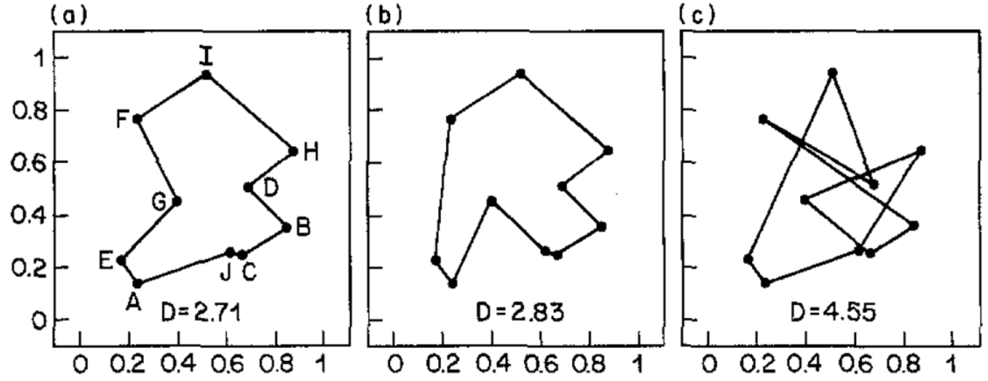


Fig. 5. A stylized picture of the syntax and connections of the TSP neural circuit. Each neuron is symbolically indicated by a square. The neurons are arranged in an n by n array. Each city is associated with n neurons in a row, and each position in the final tour is associated with n neurons in a column. A given neuron ($V_{X,j}$) represents the hypothesis that city X is in position j in the solution. The patterns of synaptic connection for two different neurons are indicated.

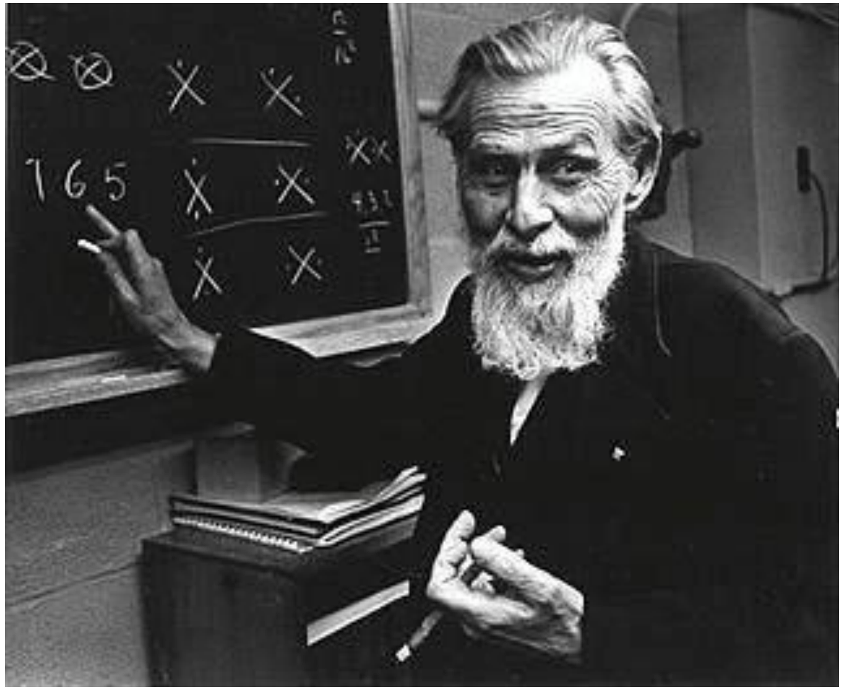


Hopfield & Tank (1985, 1986)

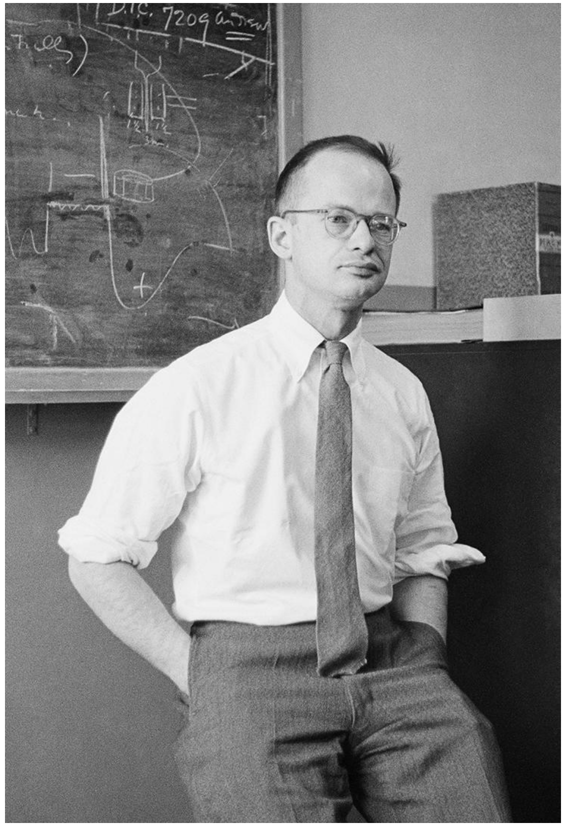
2. Single neuron dynamics

How do neurons in the Hopfield network update their activity?

McCullough-Pitts neuron model



Warren McCulloch (neuroscientist)



Walter Pitts (logician)

binary neuron operating on discrete time steps

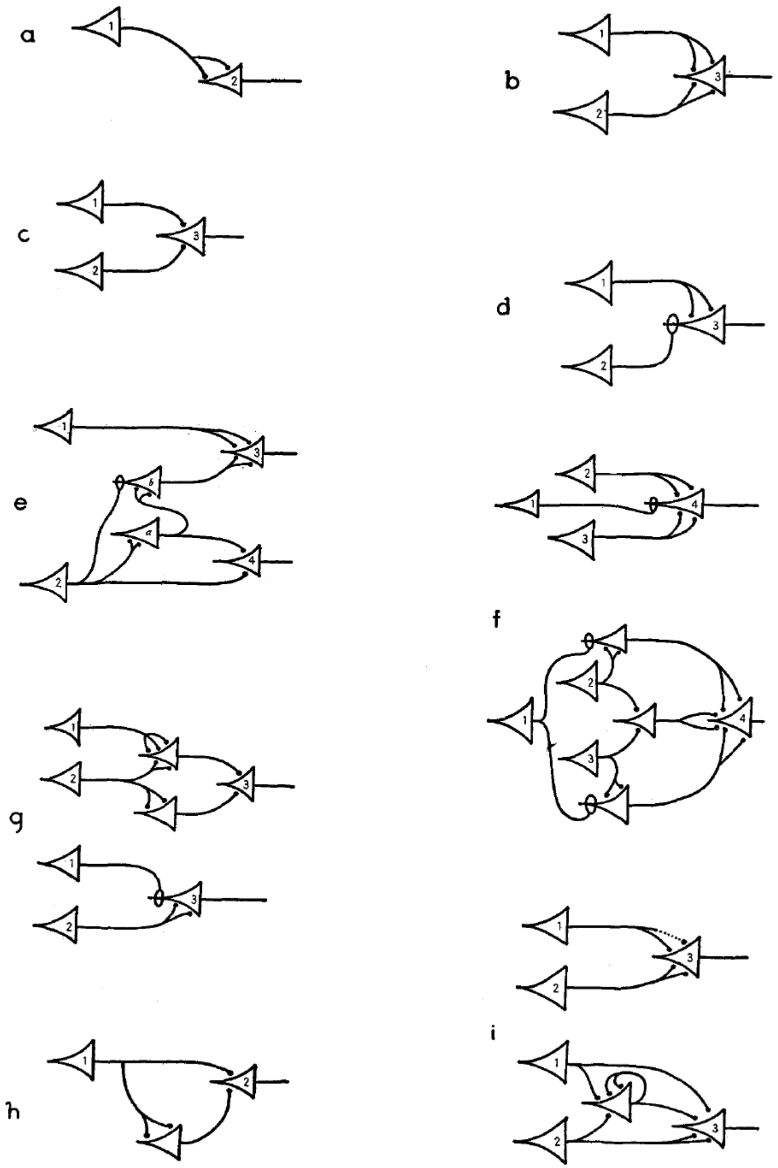


FIGURE 1

Figure 1a $N_2(t) \,.\equiv.\, N_1(t-1)$

Figure 1b $N_3(t) \,.\equiv.\, N_1(t-1) \vee N_2(t-1)$

Figure 1c $N_3(t) \,.\equiv.\, N_1(t-1).N_2(t-1)$

Figure 1d $N_3(t) \,.\equiv.\, N_1(t-1). \sim N_2(t-1)$

Figure 1e $N_3(t) \,:\equiv:\, N_1(t-1) \,.\vee.\, N_2(t-3). \sim N_2(t-2)$

$N_4(t) \,.\equiv.\, N_2(t-2).N_2(t-1)$

Figure 1f $N_4(t) \,:\equiv:\, \sim N_1(t-1).N_2(t-1) \vee N_3(t-1) \,.\vee.\, N_1(t-1).N_2(t-1).N_3(t-1)$

$N_4(t) \,:\equiv:\, \sim N_1(t-2).N_2(t-2) \vee N_3(t-2) \,.\vee.\, N_1(t-2).N_2(t-2).N_3(t-2)$

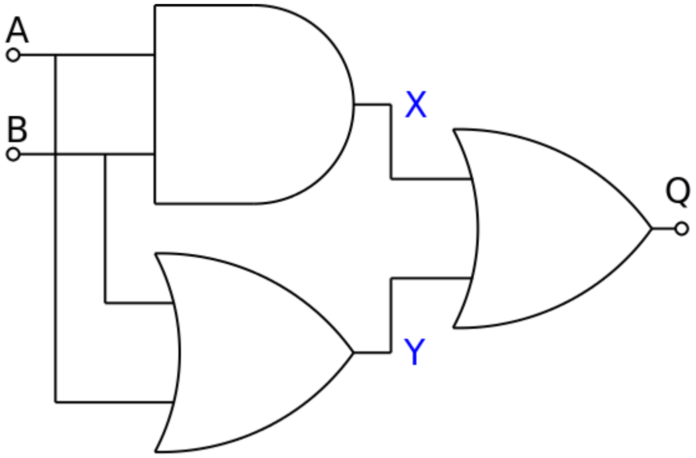
Figure 1g $N_3(t) \,.\equiv.\, N_2(t-2). \sim N_1(t-3)$

Figure 1h $N_2(t) \,.\equiv.\, N_1(t-1).N_1(t-2)$

Figure 1i $N_3(t) \,:\equiv:\, N_2(t-1) \,.\vee.\, N_1(t-1).(Ex)t-1.N_1(x).N_2(x)$

McCulloch & Pitts (1943)

Logic circuits

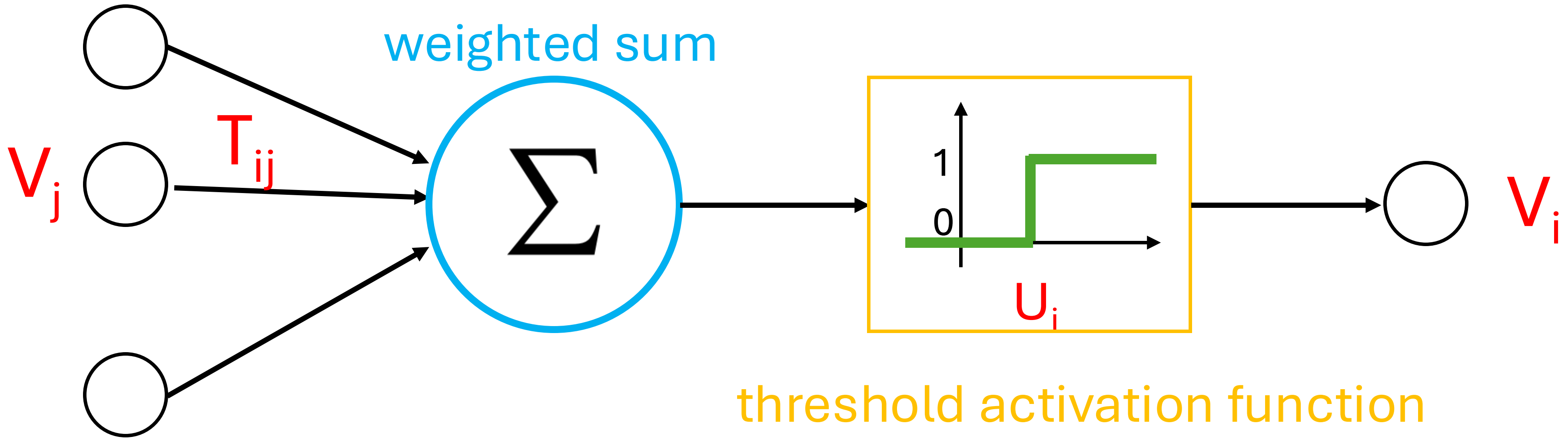


A	B	X	Y	Q
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	1

Activity of a single neuron

$$
\begin{array}{l} V_i \to 1 \\ V_i \to 0 \end{array} \quad \text{if} \quad \sum_{j \neq i} T_{ij} V_j \quad \begin{array}{l} > U_i \\ < U_i \end{array}
$$

eq (1) of the paper

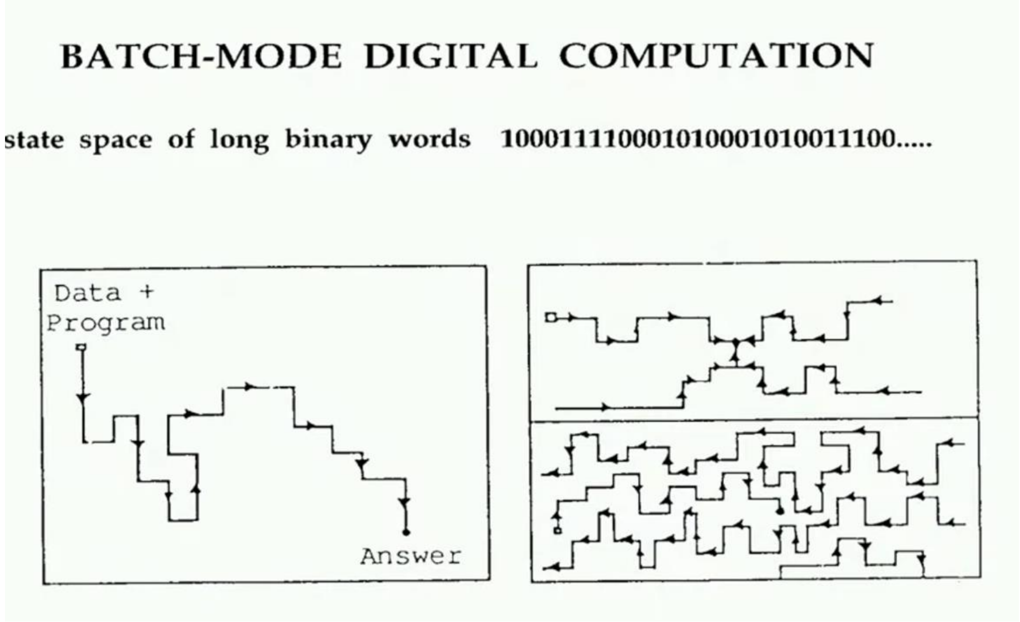


Note that this is drawn as a feedforward network, but Hopfield network is a recurrent neural network.

Computation -> dynamics in the state space

$$
\begin{array}{l}
\text{neuron 1} \\ \\ \\ \\ \\ \\ \text{neuron N}
\end{array}
\quad
\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}
\quad
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}
\quad
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}
$$

N-dimensional state space



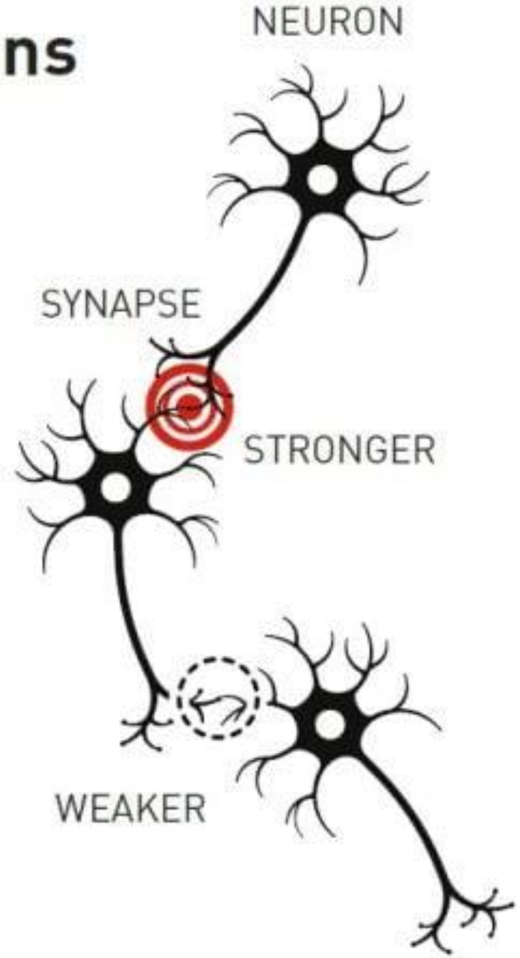
Hopfield Nobel Lecture

3. Memory formation

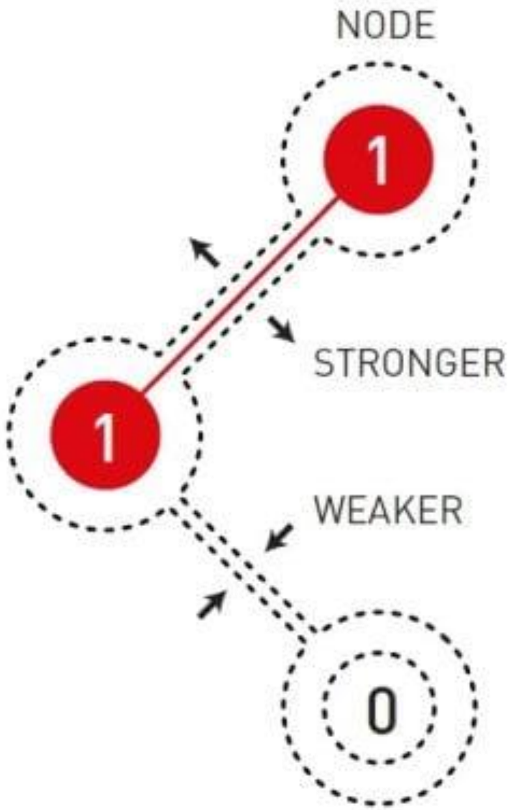
How does a Hopfield network store memories?

Natural and artificial neurons

The brain's neural network is built from living cells, neurons, with advanced internal machinery. They can send signals to each other through the synapses. When we learn things, the connections between some neurons get stronger, while others get weaker.



Artificial neural networks are built from nodes that are coded with a value. The nodes are connected to each other and, when the network is trained, the connections between nodes that are active at the same time get stronger, otherwise they get weaker.



How to store a single memory?

$$T_{ij} = (2V_i - 1)(2V_j - 1)$$

V_i	V_j	$2V_i - 1$	$2V_j - 1$	T_{ij}
0	0	-1	-1	+1
0	1	-1	+1	-1
1	0	+1	-1	-1
1	1	+1	+1	+1

Correlations between neurons are recorded in the weight matrix T.

+1

-1

$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

How to store multiple memories?

$$T_{ij} = \sum_{s} (2V_i^s - 1)(2V_j^s - 1)$$

s = 1, 2,... : memory patterns

- Linear superposition!
- Distributed memory (more robust)
- New memories can be continuously added

4. Energy function

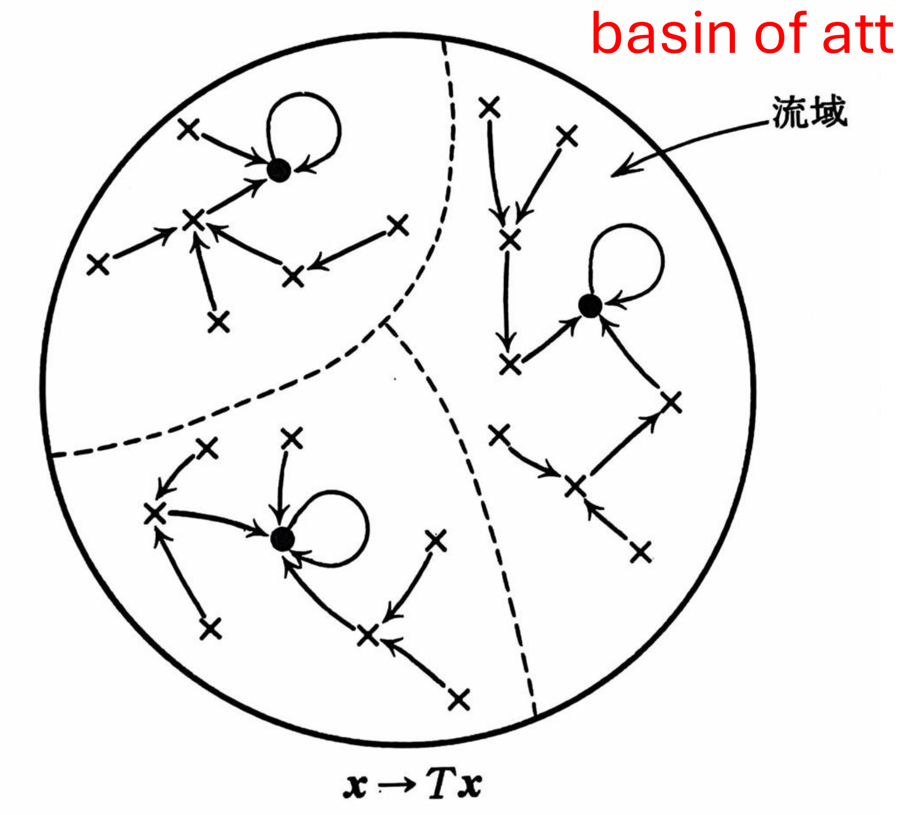
How can we describe the behavior of all the neurons?

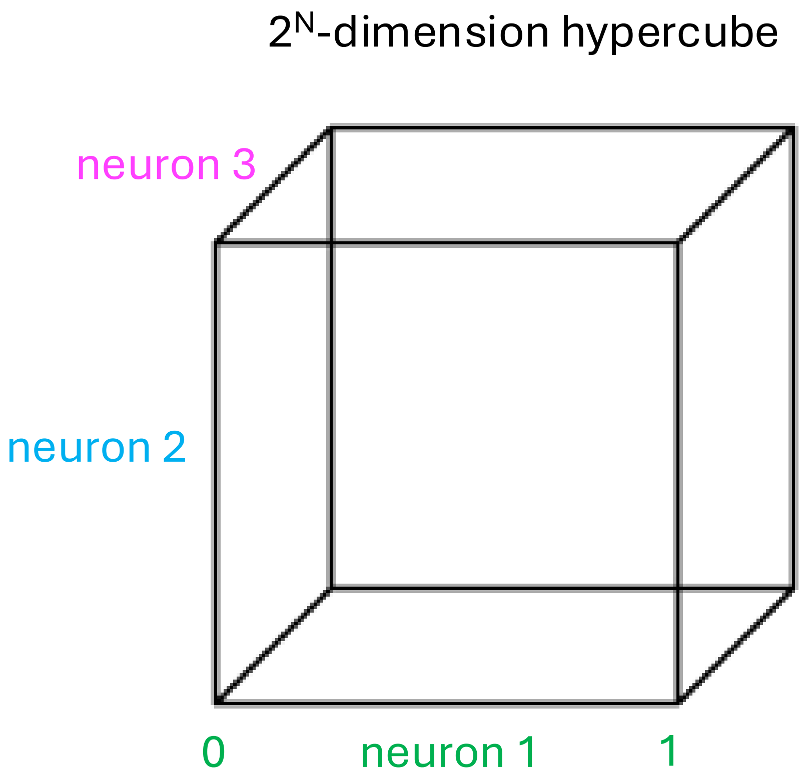
Energy summarizes the network state into a single number

$$E = -\frac{1}{2} \sum_{i \neq j} \sum T_{ij} V_i V_j$$

- “quadratic form”
- Energy is a scalar value
- Sometimes called the Lyapunov function

State transition diagram

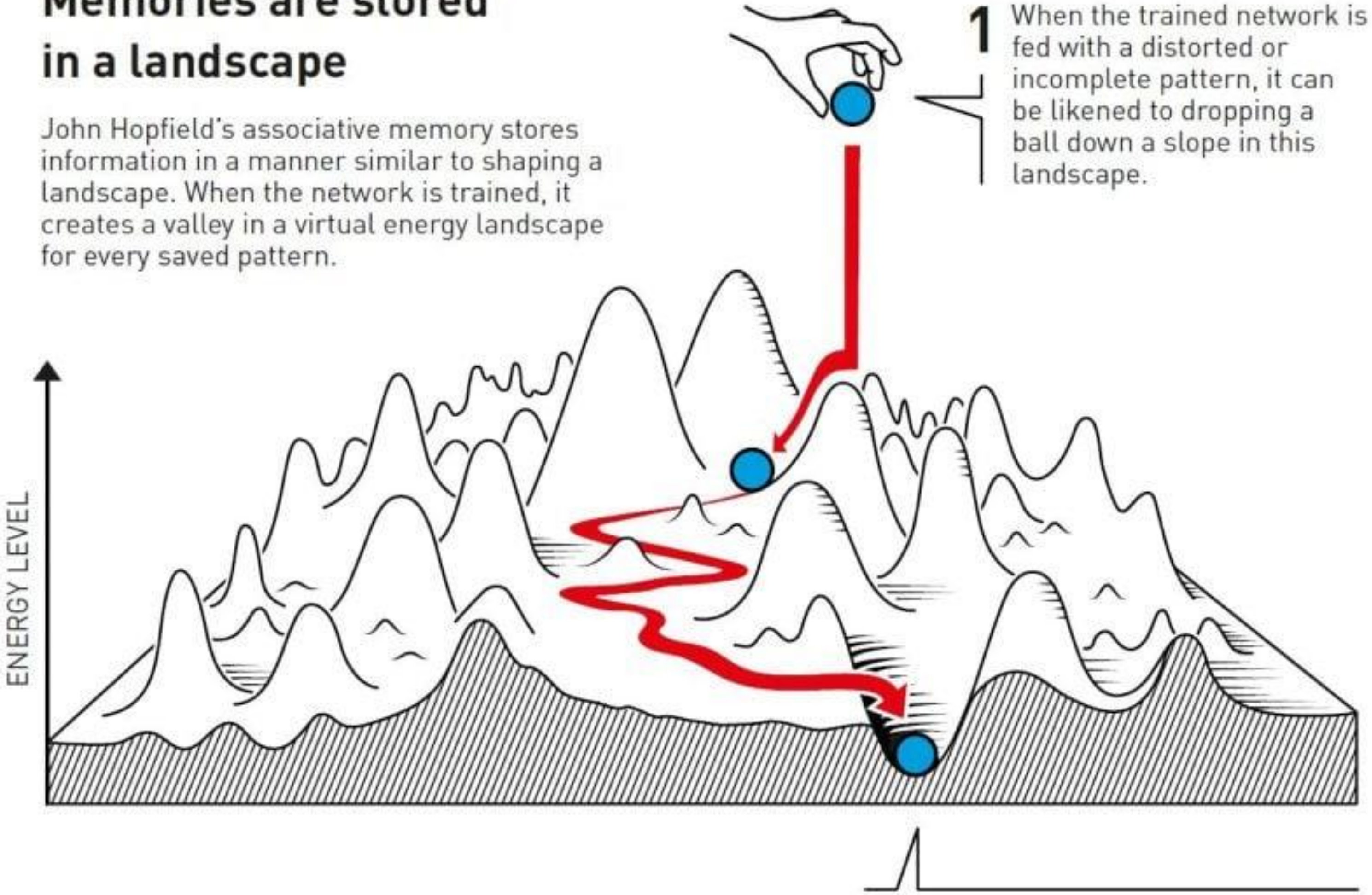




● stable state (attractor state)

Memories are stored in a landscape

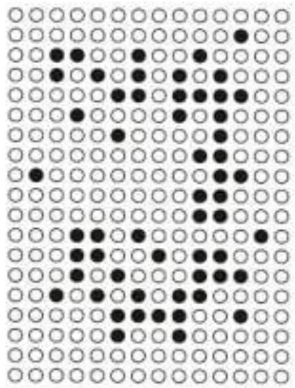
John Hopfield's associative memory stores information in a manner similar to shaping a landscape. When the network is trained, it creates a valley in a virtual energy landscape for every saved pattern.



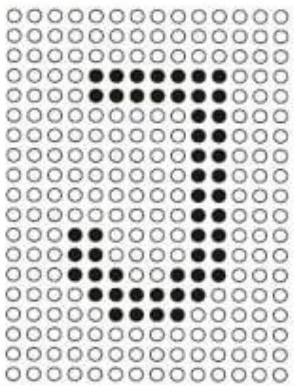
1 When the trained network is fed with a distorted or incomplete pattern, it can be likened to dropping a ball down a slope in this landscape.

2 The ball rolls until it reaches a place where it is surrounded by uphills. In the same way, the network makes its way towards lower energy and finds the closest saved pattern.

INPUT PATTERN



SAVED PATTERN

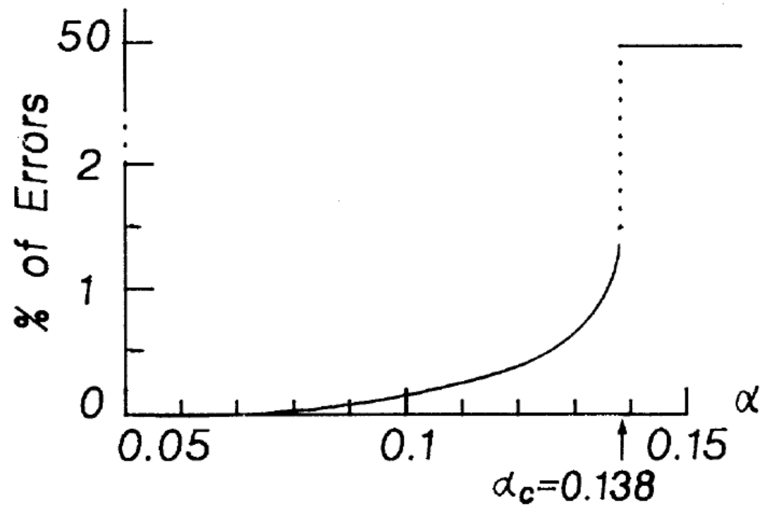


5. Memory capacity

How many different patterns can this network memorize?

Memory capacity

- How many different memory patterns can it store?
- Hopfield simulated with random patterns.
- # of memories ~ 0.15 N
- Linear in the number of neurons N



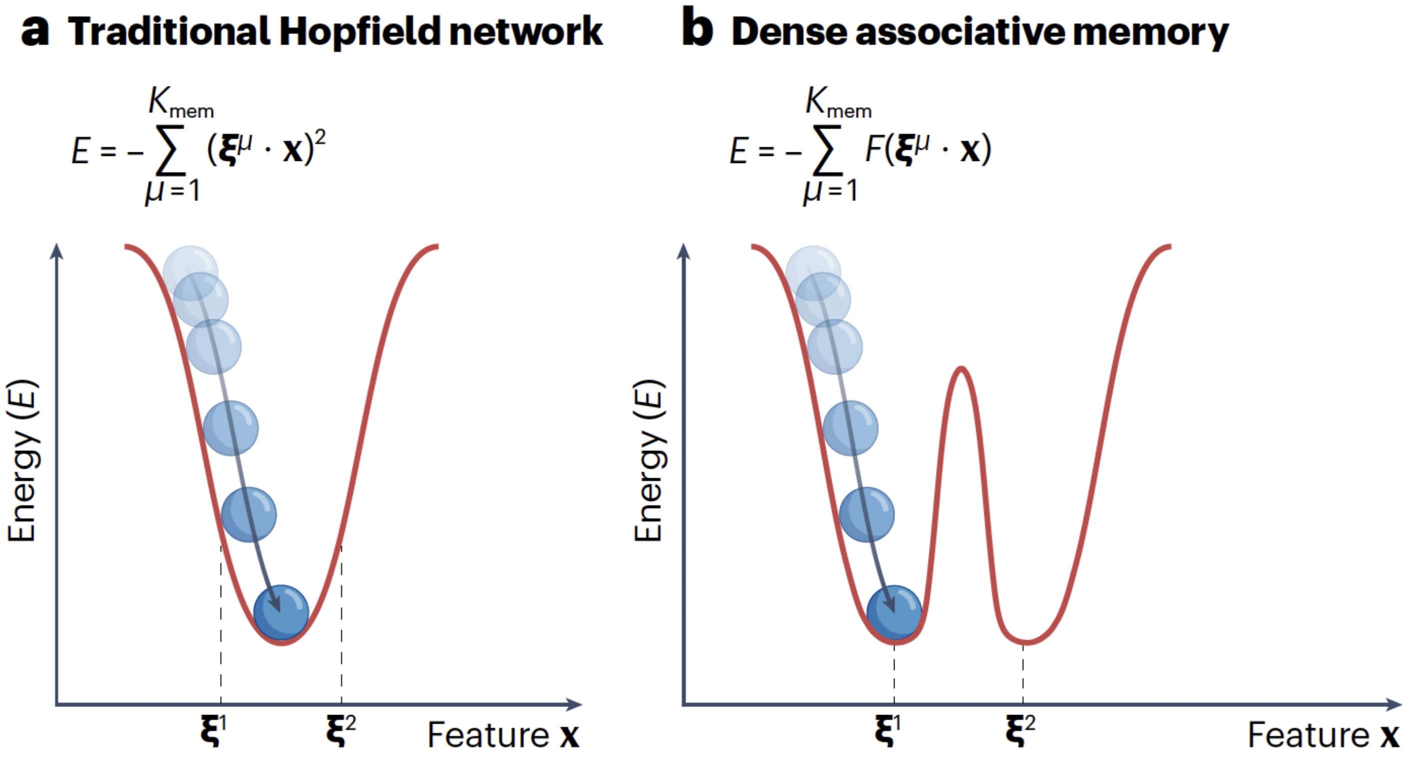
Amit…Somplinsky (1985): based on techniques used in physics

Modern Hopfield network

- Problem
  - memory capacity is linear in # of neurons

- Solution
  - use a different activation function

Modern Hopfield network



Krotov (Nat Rev Physics, 2023)

HOPFIELD NETWORKS IS ALL YOU NEED

Hubert Ramsauer* **Bernhard Schäfl*** **Johannes Lehner*** **Philipp Seidl***
Michael Widrich* **Thomas Adler*** **Lukas Gruber*** **Markus Holzleitner***
Milena Pavlović[‡,§] **Geir Kjetil Sandve**[§] **Victor Greiff**[‡] **David Kreil**[†]
Michael Kopp[†] **Günter Klambauer*** **Johannes Brandstetter*** **Sepp Hochreiter**[*,†]

*ELLIS Unit Linz, LIT AI Lab, Institute for Machine Learning,
Johannes Kepler University Linz, Austria
[†] Institute of Advanced Research in Artificial Intelligence (IARAI)
[‡] Department of Immunology, University of Oslo, Norway
[§] Department of Informatics, University of Oslo, Norway

2021

Hopfield Energy		New Energy		Update Rule		Transformer
$-\exp\left(\mathrm{lse}\left(1, \boldsymbol{\xi}^T \boldsymbol{X}\right)\right)$	$=$	$-\mathrm{lse}(\beta, \boldsymbol{\xi}^T \boldsymbol{X}) + \frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi} + c$	$\Rightarrow$	$\mathrm{softmax}\left(\beta\ \boldsymbol{\xi}^T \boldsymbol{X}\right) \boldsymbol{X}^T$	$=$	$\mathrm{softmax}\left(\frac{1}{\sqrt{d_k}}\ \boldsymbol{Q}\boldsymbol{K}^T\right) \boldsymbol{V}$

Questions

- Are all the memory states stable?
- Are there any other stable states that are not part of the memory?

- What if weight matrices are not symmetric?
- Why can we not include self-connections?
- How to make the memories orthogonal?
- What is neurons are continuous? (Hopfield 1984; Amari, Grossberg)

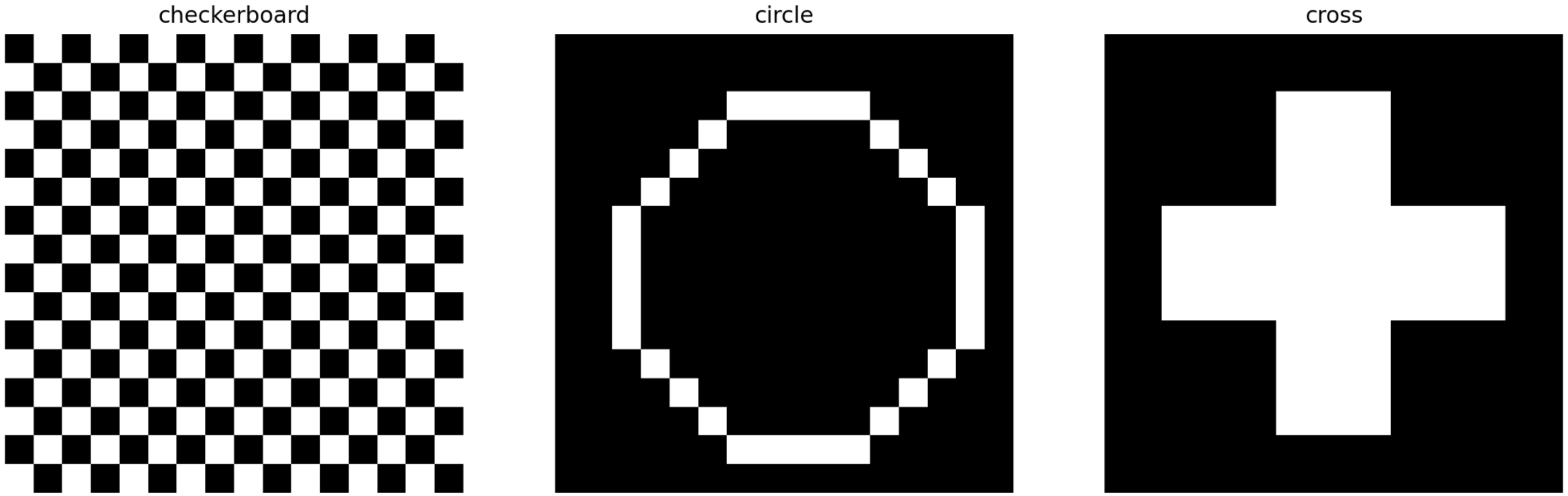
Excercises

< 10 lines of code to fill!

https://github.com/CIBR-Okubo-Lab/Amari_Hopfield

Preparation

- need matplotlib, numpy, (scikit-image)
- Just download three images in `data` folder.
  - otherwise run `generate_images.py`
  - This will create three images and save as a Numpy file (`data/images.npz`)

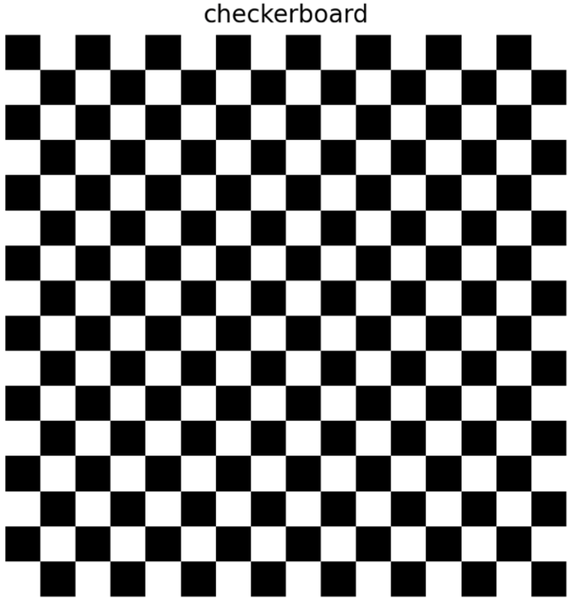


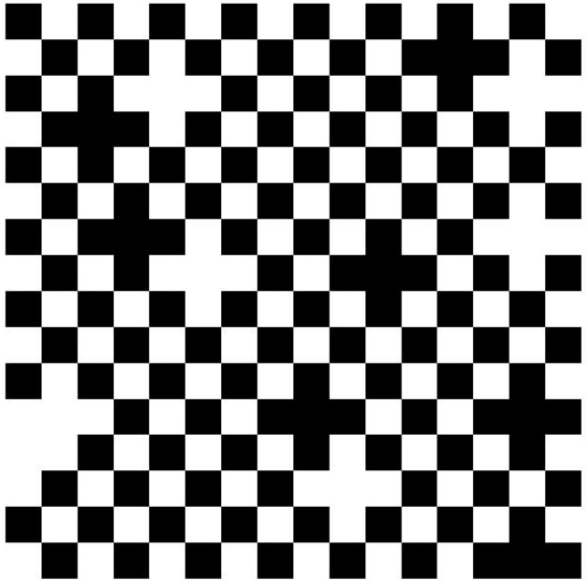
Overview of the exercise

- We will simulation asynchronous state update with a network of 256 neurons.
- Just add lines to `hopfield_exercise.py` wherever specified
- All the convenience functions are in `utils.py`

Exercise 1: memorize one pattern

- image_list = ['checkerboard']
- input_image = 'checkerboard’





Exercise 2: mutiple patterns

- image_list = ['checkerboard', 'circle', 'cross’]
- input_image = ‘cross’

  - Does the network sometimes converge to a non-memorized pattern?
  - Do all the pattern have similar “basin of attraction”?