

INTRODUCTION TO HOPFIELD NEURAL NETWORKS

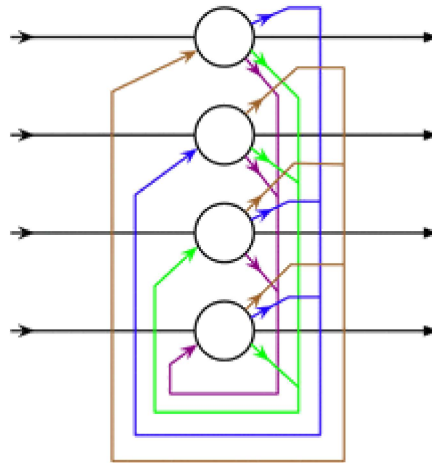
BY SIXTE DE MAUPEOU

A Hopfield⁽²⁾ neural network is a type of artificial neural network invented by John Hopfield in 1982. It usually works by first learning a number of binary patterns and then returning the one that is the most similar to a given input.

WHAT DEFINES A HOPFIELD NETWORK

It is composed of only one layer of nodes or units each of which is connected to all the others but not itself. It is therefore a feedback network, which means that its outputs are redirected to its inputs. Every unit also acts as an input and an output of the network. Thus the number of nodes, inputs, outputs of the network are equal. Additionally, each one of the neurons in a has a binary state or activation value, usually represented as 1 or -1, which is its particular output. The state of each node generally converges, meaning that the state of each node becomes fixed after a certain number of updates.

AN EXAMPLE OF A HOPFIELD NEURAL NETWORK WITH 4 NODES:



HOW A HOPFIELD NETWORK IS UPDATED

The nodes of a Hopfield network can be updated either synchronously or asynchronously. In the case of a synchronously updated network, a clock coordinates the different nodes so that they are all updated at the same time. In the case of an asynchronously updated network the nodes update one at a time and the node is selected randomly. Since there doesn't exist any clock in biological systems such as the brain to synchronise the update of the nodes, the asynchronous networks are considered to be more realistic. In both cases there exists a connectivity weight between any two distinct nodes since all nodes are connected. The weight between any given node and itself is considered to be 0. We note the connectivity weight between node n and node m $w_{nm} = w_{mn}$. This connectivity weight determines how much impact the state of node j will have on the new state of node i when node i is updated and vice-versa. If the weight between two nodes is positive the states of the two nodes will converge, otherwise they will diverge. The weight can be considered as an indicator of the final state of a node with respect to the finite state of the

others Every node i also has a threshold θ_i that determines whether or not the input is sufficient to give it a state of 1. If the total input of a node is greater than or equal to its threshold then the node's new state is 1, otherwise it is -1.

THE INPUT RECEIVED BY A NODE i , ALSO CALLED WEIGHTED INPUT SUM OF NODE i CAN BE DEFINED AS:

$$I_i = \sum_{j=1}^n w_{ij} s_j$$

Where:

- w_{ij} is the connectivity weight between i and j .
- s_j is the state of node j .

AND THE NEW STATE OF NODE i IS THEN GIVEN BY:

$$s_i = \text{sign}(I_i - \theta_i)$$

Where:

- $\text{sign}(x) = 1 \quad \forall x \geq 0$
- $\text{sign}(x) = -1 \quad \forall x < 0$

We can also describe the update of a node in terms of matrices:

THE RESPECTIVE STATES OF ALL NEURONS CAN BE EXPRESSED AS A ROW MATRIX S :

$$S = (s_1 \quad s_2 \quad \dots \quad s_i \quad \dots \quad s_n)$$

ALL THE CONNECTIVITY WEIGHTS OF A HOPFIELD NETWORK CAN BE REPRESENTED IN A WEIGHT MATRIX W :

$$W = \begin{pmatrix} 0 & w_{12} & \dots & w_{1i} & \dots & w_{1n} \\ w_{21} & 0 & \dots & w_{2i} & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ w_{i1} & w_{i2} & \dots & 0 & \dots & w_{in} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{ni} & \dots & 0 \end{pmatrix}$$

The weight matrix of a Hopfield network is always a symmetric, zero-diagonal matrix. This follows from the fact that no neuron is connected to itself and the fact that $w = w$. It also implies that neuron i influences on the new value of neuron j as much as neuron j influences on the new value of neuron i . It also means that the state of a neuron at some time t does not influence its state at time $t+1$.

THE THRESHOLDS OF ALL NEURONS FORM A COLUMN MATRIX T :

$$T = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_i \\ \vdots \\ \theta_n \end{pmatrix}$$

TRAINING A HOPFIELD NETWORK

There exist several rules to train a Hopfield network all of which are based on the same process: the modification of the synaptic weights. Given an input to learn, the connectivity weight between nodes with the same state is increased whereas the weight between nodes with opposite states is decreased. A learning rule can have two useful properties: being local and being incremental. It is local if for each update it uses only the information from both nodes whose connectivity weight is updated. It is

incremental if it does not require any information about the other patterns previously learnt by the network. Another major property of a learning rule is the capacity. It corresponds to the number of patterns per neuron that can be recalled by the network. The most common learning rule for Hopfield networks is Hebbian⁽³⁾ learning, invented by Donald Hebb in 1949.

FOR A NETWORK WITH N NODES HEBBIAN LEARNING UPDATES THE CONNECTIVITY WEIGHTS ACCORDING TO THE FORMULA:

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^P \epsilon_i^{\mu} \epsilon_j^{\mu}$$

Where:

- ϵ_x^{μ} is the state of node x in pattern μ

As the formula describes it, Hebbian learning consists of setting the new weight between nodes i and j to the average of the product of the states of i and j in each of the patterns. While the Hebbian learning rule is the most well-known, the Storkey learning rule, invented by Amos Storkey in 1997, is also both local and incremental and provides a network with a greater capacity.

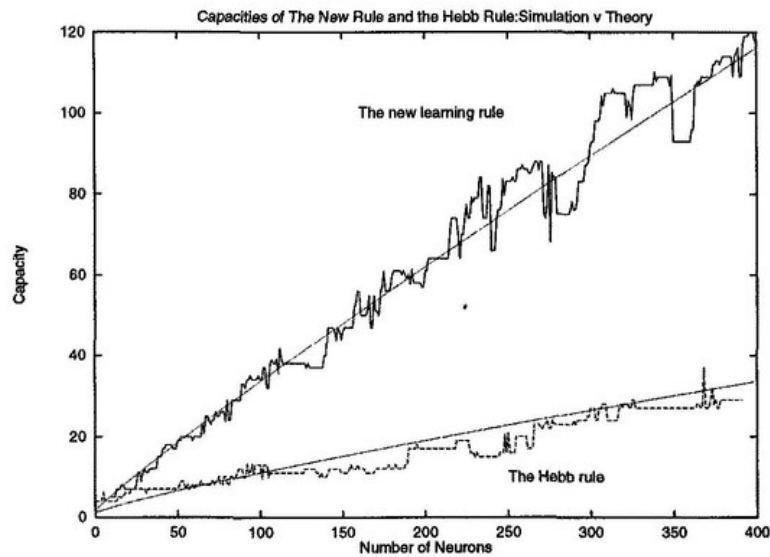
THE STORKEY LEARNING RULE WAS MATHEMATICALLY DEFINED BY AMOS STORKEY AS:

$$w_{ij}^0 = 0 \quad \forall i, j \quad \text{and} \quad w_{ij}^{\nu} = w_{ij}^{\nu-1} + \frac{1}{n} \xi_i^{\nu} \xi_j^{\nu} - \frac{1}{n} \xi_i^{\nu} h_{ji}^{\nu} - \frac{1}{n} h_{ij}^{\nu} \xi_j^{\nu}$$

Where:

- w_{ij}^{ν} represents the weight between i and j after the ν^{th} pattern has been learnt.
- ξ^{ν} is the new learning pattern.
- $h_{ij}^{\nu} = \sum_{k=1, k \neq i, j}^n w_{ik}^{\nu-1} \xi_k^{\nu}$ is a form of local field.⁽¹⁾

THE FOLLOWING GRAPH PUBLISHED BY STORKEY SHOWS THE DIFFERENCE IN CAPACITY BETWEEN THE HEBBIAN LEARNING AND THE STORKEY LEARNING:



ACCORDING TO THE MODEL DEDUCED FROM THE ABOVE GRAPH, FOR N NODES WE HAVE:

HEBBIAN LEARNING ABSOLUTE CAPACITY

$$\frac{N}{2\ln(N)}$$

STORKEY LEARNING ABSOLUTE CAPACITY

$$\frac{N}{2\sqrt{\ln(N)}}$$

THE ENERGY OF A HOPFIELD NETWORK

IN A HOPFIELD NETWORK EACH NODE HAS AN ASSOCIATED ENERGY WHICH IS DEFINED AS:

$$E_i = -\frac{1}{2}I_i s_i$$

Where:

- I_i is the weighted input sum of node i
- s_i is the state of i

FROM THIS EXPRESSION WE CAN DEDUCE THE TOTAL ENERGY OF A NETWORK WITH N NODES:

$$E = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ij} s_i s_j + \sum_{i=1}^n \theta_i s_i$$

THIS EXPRESSION CAN NOW BE CLARIFIED USING THE PREVIOUSLY DEFINED MATRICES:

$$E = -\frac{1}{2} S W S^T + T^T S^T$$

If a neuron changes sign when updated then the energy of the network decreases otherwise it stays constant. Therefore the energy indicates whether or not the network is modified when an update occurs and reaches its minimum energy when it becomes stable. The training patterns all correspond to a local minimum, called attractor on the graph of the energy, thus once the network reaches a training pattern it stops evolving. However some local minima may not correspond to a training pattern, these are called spurious minima and constitute an obvious limit to the Hopfield network.

