Hopfield network

A **Hopfield network** is a recurrently connected network; it is intended to perform pattern completion and was proposed by John Hopfield in 1982 [1], though other people had had the idea before in different contexts. The idea behind a Hopfield network is that you evolve the network according to the McCulloch-Pitts relation, so, in the synchronous update version, from one iteration to the next

$$\hat{x}_i = \phi\left(\sum_j w_{ij} x_j\right) \tag{1}$$

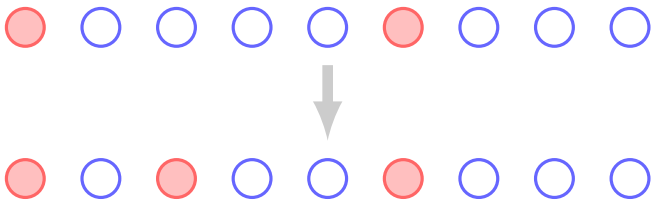
and then $x_i \longrightarrow \hat{x}$; that is all the nodes update using the old values. In the most common version of a Hopfield network, the w_{ij} are symmetric, that is

$$w_{ij} = w_{ji} \tag{2}$$

The threshold values θ_i have been set to zero, this is something you can do in a Hopfield network if you want because the learning rule doesn't change to threshold. In the asynchronous scheme, the you update the nodes one-by-one, for example, after choosing a random node; for simplicity we stick to synchronous updates here.

The idea is that this is a model for **auto-associative** memory. Auto-associative memories are patterns representing memories along with some dynamics that complete partial patters. Imagine a sequence of McCulloch-Pitts neurons

where the filled circles correspond to on. Recall occurs when the network is presented with a partial pattern and evolves into the complete patterns.



The Hopfield network is intended to model the recall, and learning, of memories in an autoassociative network.

One way to think about this is to note that there is an 'energy' associated with a Hopfield network:

$$E = -\frac{1}{2}\sum_{ij} w_{ij}x_i x_j \tag{3}$$

and you can show than if you update a node you will reduce the energy. Roughly speaking if you update a node x_i then it is more likely to have the same sign as a connected node x_j if the connection between them is large and positive since this means $w_{ij}x_j$ will have a big effect on the activation of x_i and vice versa. x_i and x_j are more likely to have an opposite sign if the connection is large and negative. Thus, updating will tend to make terms like $w_{ij}x_i x_j$ into a positive number if w_{ij} is large and so that updating the neurons will tend to reduce the energy. In fact, this can be proved and that the system will evolve to a local minimum.

Now, if the dynamics reduces the energy, E, the goal is to pick the connection strenghts so that the minima of E correspond to the patterns that the network is charged with storing. Now the question is how to create the correct local minima? Here is a rules to achieve this:

$$w_{ij} = \frac{1}{N}\sum_{a} x_i^a x_j^a \tag{4}$$

where N is the number of patterns to be stored, and a indexes the patterns. There are other rules, in fact there are rules that can store more patterns, but this rule, a sort of 'top down' rule is inspired by Hebbian plasticity. There is an online rule that is even closer to Hebbian plasticity where w_{ij} is changed for each presentation:

$$\Delta w_{ij} = \frac{\eta}{4}(x_i^a + 1 - 2\alpha)(x_j^a + 1 - 2\alpha) \tag{5}$$

Following this online rule will bring you to the values in the formula for creating the correct minima: Eqn. 4. The α's are an ofset value which allow the network to reach a stable equilibrium since Δw_{ij} should stop changing (on average) if the average fraction of iterations where x_i or x_j is plus one is equal to α.

These changes in the w_{ij} mimick a simple version of **Hebbian plasticity**, a putative description of how the synapses in the brain change in response to

neuronal activity. Synaptic plasticity usually refers to the long-term changes in synapse strength, a long term increase in synaptic strength is called **long term potentiation** of LTP, a decrease is called **long term depression** or LTD. It is believed that synapses respond to their pre- and post-synaptic activity, so that the changes depend on the behavior of the pre- and post-synaptic neurons. It is not known in detail what rules govern this plasticity, it seems different neurons have different plasticity rules.

The closest thing to an overall rule was formulated by Hebb in 1949 when he said [2]:

> Let us assume that the persistence or repetition of a reverberatory activity (or 'trace') tends to induce lasting cellular changes that add to its stability. [. . .] When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

In other words, if one neurons tends to cause another to fire, the synapse from the first to the second will get stronger. In artificial neurons or rate-based neurons, lack spiking dynamics and instead have a continuous state or rate variable; since **Hebbian plasticity** often plays a role in artificial neural networks it is often applied to a rule that strengthens synapses between neurons that are active at the same time, that is, the explicit causal structure is ignored in favor of

> Neurons that fire together wire together.

This leads to a plasticity rule

$$\Delta w_{ij} = \eta x_i x_j \tag{6}$$

where w_{ij} is the strength of the synapse from neuron i to neuron j, x_i and x_j are the states of the two neurons and η is a learning rate. Another version is

$$\Delta w_{ij} = \eta(x_i - \alpha)(x_j - \alpha) \tag{7}$$

where having α allows for different cut-off points between the behaviour that causes potentiation or depression.

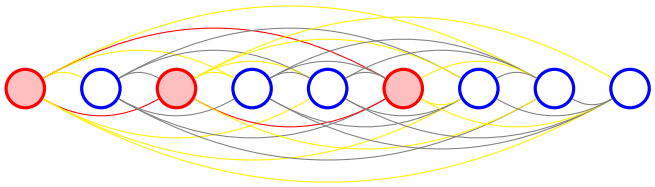


Figure 1: Learning in the associate network. The pattern has been imposed and connection strengths are changed. The red links increase by $\eta(1-\alpha)^2$ and the gray by $\eta(-\alpha)^2$, the yellow links decrease by $\eta\alpha(1-\alpha)$.

This is clearly, up to a change in variables for the α and rescaling of η, this is the same as the rule

$$\Delta w_{ij} = \frac{\eta}{4}(x_i^a + 1 - 2\alpha)(x_j^a + 1 - 2\alpha) \tag{8}$$

mentioned for Hopfield networks; in fact, in the Hopfield network α has to be set equal to the average density of the patterns, that is, the average number of ones, for convergence.

So, to recap, during learning the patterns are activated and plastic changes are made to the synapse strength according to a simple correlation based Hebbian plasticity rule. In the brain, the idea is that outside activity, signals from outside the auto-associative network, will cause some of the neurons, the pattern to be learned, to be active while others remain dormant. During this time plasticity occurs. Later, during recall, the outside activity causes a fraction of the pattern to become activity. The internal dynamics of the network, the McCulloch-Pitts updates, cause other neurons to also become active, allowing the pattern to repeat.

Typically α is very small for real networks so there will be a large increase for the connection between two neurons that are active at the same time, a tiny increase for pairs neurons that are inactive at the same time and a medium size decrease for pairs of neurons where one is active and one inactive. See Fig. 1.

As discussed, during recall some of the neurons are held in the active state and the rest of the network evolves according to a threshold input rule. That

means each neuron has an input given by

$$r_i = \sum w_{ij} x_j \tag{9}$$

and is set in the active state if $r_i > 0$. The idea is that after learning the pattern $\{0, 2, 5\}$

the connections between these nodes will be strong, so if the network has nodes $\{0, 5\}$ activated

the value $r_2 = w_{12} + w_{52}$ will be larger than the threshold and the subsequent dynamics will switch neuron 2 on. However, in this network, if a different initial set of neurons are activated, the activity will die away because the r_i will all be sub-threshold.

When many patterns are stored it is likely that there will be interference between them. This is illustrated in Fig. 2. Although the figure shows how a single neuron fails to participate in two patterns, for larger networks some overlap is possible, but too much overlap prevents retrieval. In fact the capacity is proportional to the number of neurons, N. A hand-waving argument goes like this: the number of connections is roughly N^2 and the amount of information in a pattern is N so the number of patterns that can be stored is $N^2/N = N$ [3].

The capacity is also larger if there is sparseness; one way to think of this is to observe the weight decrease between an active and inactive connection is $\Delta w_{ij} = -\eta\alpha(1-\alpha)$ so the smaller α is the smaller the amount these links are decreased. Links are decreased if, in the pattern, one neuron is active and one inactive, they are strengthened if both neurons are active, the increase is $\eta(1-\alpha)^2$. Hence, it takes of the order of $1/\alpha$ patterns where a connection is weakened to wipe out the strengthening that results if the connection is part of a pattern. In fact, it is estimated that the capacity of a network is

$$P = \frac{k}{\alpha} N \tag{10}$$

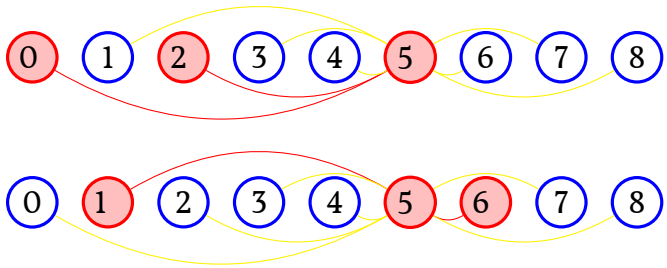


Figure 2: Interference in an auto-associative network. Neuron 5 is involved in two patterns and, as a consequence, some of its connections are strengthen for one pattern and weakened for the other, if these strengthening and weakening effects are similar in size it makes it unlikely that either pattern will be accurately retrieved.

where k is constant which has been found to be about $k \approx 0.035$, this is reduced to

$$P = c\frac{k}{\alpha}N \tag{11}$$

if there are missing connections, where c stands for the fraction of pairs that are connected.

The actual sparseness of the brain needs to balance this advantage, the increased capacity, along with a metabolic advantage and more abstract computational advantage which says that a sparse coding for information involves object recognition or segmentation against the disadvantages, most obviously the vulnerability of the pattern to the loss of neurons or connections and, perhaps more importantly, a sparse code involves fewer elements and so may be less useful for retrieval. It is hard to actually estimate sparseness in practice since neurons are not, in reality, on-off units.

References

[1] Hopfield, JJ. (1982) Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the USA, 79:2554–2558.

[2] Hebb DO. (1949) The Organization of Behavior. New York: Wiley & Sons.

[3] Amit D. (1992) Modeling Brain Function: The World of Attractor Neural Networks. Cambridge University Press, Cambridge England.