



WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction

[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools

[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

Print/export

[Create a book](#)
[Download as PDF](#)
[Printable version](#)

Languages

[العربية](#)
[Català](#)
[Deutsch](#)
[Español](#)
[فارسی](#)
[Français](#)
[Latviešu](#)
■■■■■■■
[Nederlands](#)
[日本語](#)
[Polski](#)
[Русский](#)
[Svenska](#)
[Українська](#)
[Tiếng Việt](#)
[中文](#)

[Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Hopfield network

From Wikipedia, the free encyclopedia
(Redirected from [Hopfield net](#))

A **Hopfield network** is a form of [recurrent artificial neural network](#) popularized by [John Hopfield](#) in 1982, but described earlier by Little in 1974.^{[1][2]} Hopfield nets serve as [content-addressable memory](#) systems with [binary threshold nodes](#). They are guaranteed to converge to a [local minimum](#), but convergence to a false pattern (wrong local minimum) rather than the stored pattern (expected local minimum) can occur. Hopfield networks also provide a model for understanding human memory.

Contents [hide]

- Structure
- Updating
 - 2.1 Neurons attract or repel each other
- Energy
- Initialization and running
- Training
 - 5.1 Learning rules
 - 5.2 Hebbian learning rule for Hopfield networks
 - 5.3 The Storkey learning rule
- Spurious patterns
- Capacity
- Human memory
- See also
- References
- External links

Structure [\[edit\]](#)

The units in Hopfield nets are binary threshold units, i.e. the units only take on two different values for their states and the value is determined by whether or not the units' input exceeds their threshold. Hopfield nets normally have units that take on values of 1 or -1, and this convention will be used throughout this page. However, other literature might use units that take values of 0 and 1.

Every pair of units *i* and *j* in a Hopfield network have a connection that is described by the connectivity weight *w_{ij}*. In this sense, the Hopfield network can be formally described as a complete undirected graph *G* = *<V, f>*, where *V* is a set of [McCulloch-Pitts neurons](#) and *f* : *V*² → *R* is a function that links pairs of nodes to a real value, the connectivity weight.

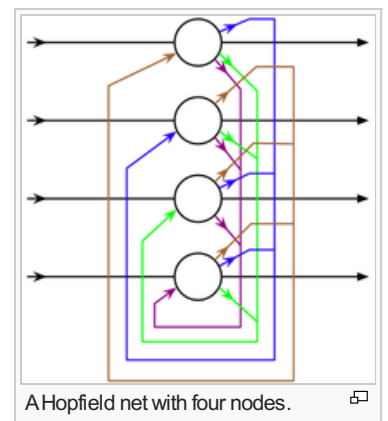
The connections in a Hopfield net typically have the following restrictions:

- w_{ii}* = 0, *∀i* (no unit has a connection with itself)
- w_{ij}* = *w_{ji}*, *∀i, j* (connections are symmetric)

The requirement that weights be symmetric is typically used, as it will guarantee that the energy function decreases monotonically while following the activation rules, and the network may exhibit some periodic or chaotic behaviour if non-symmetric weights are used. However, Hopfield found that this chaotic behavior is confined to relatively small parts of the phase space, and does not impair the network's ability to act as a content-addressable associative memory system.

Updating [\[edit\]](#)

Updating one unit (node in the graph simulating the artificial neuron) in the Hopfield network is performed using



the following rule:

$$s_i \leftarrow \begin{cases} +1 & \text{if } \sum_j w_{ij}s_j \geq \theta_i, \\ -1 & \text{otherwise.} \end{cases}$$

where:

- w_{ij} is the strength of the connection weight from unit j to unit i (the weight of the connection).
- s_j is the state of unit j .
- θ_i is the threshold of unit i .

Updates in the Hopfield network can be performed in two different ways:

- **Asynchronous:** Only one unit is updated at a time. This unit can be picked at random, or a pre-defined order can be imposed from the very beginning.
- **Synchronous:** All units are updated at the same time. This requires a central clock to the system in order to maintain synchronization. This method is less realistic, since biological or physical systems lack a global clock that keeps track of time.

Neurons attract or repel each other [\[edit\]](#)

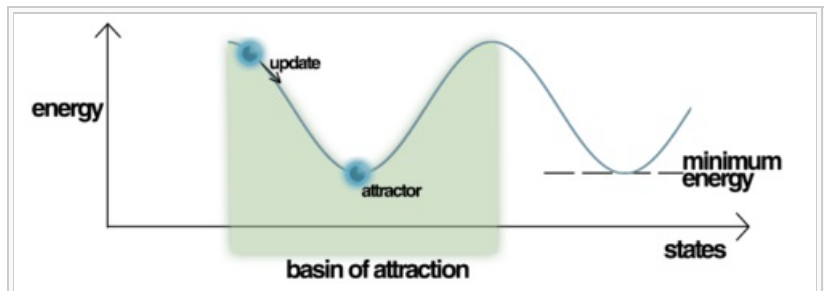
The weight between two units has a powerful impact upon the values of the neurons. Consider the connection weight w_{ij} between two neurons i and j . If $w_{ij} > 0$, the updating rule implies that:

- when $s_j = 1$, the contribution of j in the weighted sum is positive. Thus, s_i is pulled by j towards its value $s_j = 1$
- when $s_j = -1$, the contribution of j in the weighted sum is negative. Then again, s_i is pulled by j towards its value $s_j = -1$

Thus, the values of neurons i and j will converge if the weight between them is positive. Similarly, they will diverge if the weight is negative.

Energy [\[edit\]](#)

Hopfield nets have a scalar value associated with each state of the network referred to as the "energy", E , of the network, where:



Energy Landscape of a Hopfield Network, highlighting the current state of the network (up the hill), an attractor state to which it will eventually converge, a minimum energy level and a basin of attraction shaded in green. Note how the update of the Hopfield Network is always going down in Energy. [\[edit\]](#)

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

This value is called the "energy" because the definition ensures that when units are randomly chosen to update, the energy E will either lower in value or stay the same. Furthermore, under repeated updating the network will eventually converge to a state which is a [local minimum](#) in the energy function (which is considered to be a [Lyapunov function](#)). Thus, if a state is a local minimum in the energy function, it is a stable state for the network. Note that this energy function belongs to a general class of models in [physics](#), under the name of [Ising models](#); these in turn are a special case of [Markov networks](#), since the associated [probability measure](#), the [Gibbs measure](#), has the [Markov property](#).

Initialization and running [\[edit\]](#)

Initialization of the Hopfield Networks is done by setting the values of the units to the desired start pattern. Repeated updates are then performed until the network converges to an attractor pattern. Convergence is generally assured, as Hopfield proved that the attractors of this [nonlinear dynamical system](#) are stable, not periodic or chaotic as in some other systems. Therefore, in the context of Hopfield Networks, an attractor pattern is a final stable state, a pattern that cannot change any value within it under updating.

Training [\[edit\]](#)

Training a Hopfield net involves lowering the energy of states that the net should "remember". This allows the net to serve as a content addressable memory system, that is to say, the network will converge to a "remembered" state if it is given only part of the state. The net can be used to recover from a distorted input to the trained state that is most similar to that input. This is called associative memory because it recovers memories on the basis of similarity. For example, if we train a Hopfield net with five units so that the state (1, 0, 1, 0, 1) is an energy minimum, and we give the network the state (1, 0, 0, 0, 1) it will converge to (1, 0, 1, 0, 1). Thus, the network is properly trained when the energy of states which the network should remember are local minima.

- Note: In contrast to [Perceptron](#) training, the thresholds of the neurons are never updated.

Learning rules [\[edit\]](#)

There are various different learning rules that can be used to store information in the memory of the Hopfield Network. It is desirable for a learning rule to have both of the following two properties:

- *Local*: A learning rule is *local* if each weight is updated using information available to neurons on either side of the connection that is associated with that particular weight.
- *Incremental*: New patterns can be learned without using information from the old patterns that have been also used for training. That is, when a new pattern is used for training, the new values for the weights only depend on the old values and on the new pattern.^[3]

These properties are desirable, since a learning rule satisfying them is more biologically plausible. For example, since the human brain is always learning new concepts, one can reason that human learning is incremental. A learning system that would not be incremental would generally be trained only once, with a huge batch of training data.

Hebbian learning rule for Hopfield networks [\[edit\]](#)

The [Hebbian Theory](#) was introduced by Donald Hebb in 1949, in order to explain "associative learning", in which simultaneous activation of neuron cells leads to pronounced increases in synaptic strength between those cells.^[4] It is often summarized as "Neurons that fire together, wire together. Neurons that fire out of sync, fail to link".

The Hebbian rule is both local and incremental. For the Hopfield Networks, it is implemented in the following manner, when learning n binary patterns:

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

where ϵ_i^{μ} represents bit i from pattern μ .

If the bits corresponding to neurons i and j are equal in pattern μ , then the product $\epsilon_i^{\mu} \epsilon_j^{\mu}$ will be positive. This would, in turn, have a positive effect on the weight w_{ij} and the values of i and j will tend to become equal. The opposite happens if the bits corresponding to neurons i and j are different.

The Storkey learning rule [\[edit\]](#)

This rule was introduced by [Amos Storkey](#) in 1997 and is both local and incremental. Storkey also showed that a Hopfield network trained using this rule has a greater capacity than a corresponding network trained using the Hebbian rule.^[5] The weight matrix of an attractor neural network is said to follow the Storkey learning rule if it obeys:

$$w_{ij}^{\nu} = w_{ij}^{\nu-1} + \frac{1}{n} \epsilon_i^{\nu} \epsilon_j^{\nu} - \frac{1}{n} \epsilon_i^{\nu} h_{ji}^{\nu} - \frac{1}{n} \epsilon_j^{\nu} h_{ij}^{\nu}$$

where $h_{ij}^{\nu} = \sum_{k=1, k \neq i, j}^n w_{ik}^{\nu-1} \epsilon_k^{\nu}$ is a form of *local field* ^[3] at neuron i .

This learning rule is local, since the synapses take into account only neurons at their sides. The rule makes use of more information from the patterns and weights than the generalized Hebbian rule, due to the effect of the local field.

Spurious patterns [\[edit\]](#)

Patterns that the network uses for training, called *retrieval states*, become attractors of the system. Repeated

updates would eventually lead to convergence to one of the retrieval states. However, sometimes the network will converge to spurious patterns, that are different from the training patterns.^[6] The energy in these spurious patterns is also a local minima. For each stored pattern x , the negation $-x$ is also a spurious pattern.

A spurious state can also be a linear combination of an odd number of retrieval states. For example, when using 3 patterns μ_1, μ_2, μ_3 , one can get the following spurious state:

$$\epsilon_i^{mix} = \pm \text{sgn}(\pm \epsilon_i^{\mu_1} \pm \epsilon_i^{\mu_2} \pm \epsilon_i^{\mu_3})$$

Spurious patterns that have an even number of states cannot exist, since they might sum up to zero ^[6]

Capacity ^[edit]

The Network capacity of the Hopfield network model is determined by neuron amounts and connections within a given network. Therefore, the number of memories that are able to be stored are dependent on neurons and connections. Furthermore, it was shown that the recall accuracy between vectors and nodes was 0.138 (approximately 138 vectors can be recalled from storage for every 1000 nodes) (Hertz et al., 1991). Therefore, it is evident that many mistakes will occur if you try to store a large number of vectors. When the Hopfield model does not recall the right pattern, it is possible that an intrusion has taken place, since semantically related items tend to confuse the individual, and recollection of the wrong pattern occurs. Therefore, the Hopfield network model is shown to confuse one stored item with that of another upon retrieval. Perfect recalls and high capacity, >0.14 , can be loaded in the network by Hebbian learning method.^{[7][8]}

Human memory ^[edit]

The Hopfield model accounts for [associative memory](#) through the incorporation of memory vectors. Memory vectors can be slightly used, and this would spark the retrieval of the most similar vector in the network. However, we will find out that due to this process, intrusions can occur. In associative memory for the Hopfield network, there are two types of operations: auto-association and hetero-association. The first being when a vector is associated with itself, and the latter being when two different vectors are associated in storage. Furthermore, both types of operations are possible to store within a single memory matrix, but only if that given representation matrix is not one or the other of the operations, but rather the combination (auto-associative and hetero-associative) of the two. It is important to note that Hopfield's network model utilizes the same learning rule as [Hebb's \(1949\) learning rule](#), which basically tried to show that learning occurs as a result of the strengthening of the weights by when activity is occurring.

Rizzuto and Kahana (2001) were able to show that the neural network model can account for repetition on recall accuracy by incorporating a probabilistic-learning algorithm. During the retrieval process, no learning occurs. As a result, the weights of the network remains fixed, showing that the model is able to switch from a learning stage to a recall stage. By adding contextual drift we are able to show the rapid forgetting that occurs in a Hopfield model during a cued-recall task. The entire network contributes to the change in the activation of any single node.






McCullough and Pitts' (1943) dynamical rule, which describes the behavior of neurons, does so in a way that shows how the activations of multiple neurons map onto the activation of a new neuron's firing rate, and how the weights of the neurons strengthen the synaptic connections between the new activated neuron (and those that activated it). Hopfield would use McCullough-Pitts's dynamical rule in order to show how retrieval is possible in the Hopfield network. However, it is important to note that Hopfield would do so in a repetitious fashion. Hopfield would use a nonlinear activation function, instead of using a linear function. This would therefore create the Hopfield dynamical rule and with this, Hopfield was able to show that with the nonlinear activation function, the dynamical rule will always modify the values of the state vector in the direction of one of the stored patterns.

See also ^[edit]









- [Associative memory \(disambiguation\)](#)
- [Autoassociative memory](#)
- [Boltzmann machine](#) – like a Hopfield net but uses annealed Gibbs sampling instead of gradient descent
- [Dynamical systems model of cognition](#)
- [Ising model](#)
- [Hebbian theory](#)

References ^[edit]

- [^] Gurney, Kevin (2002). *An Introduction to Neural Networks*. Routledge. ISBN 1857285034.

2. [^] ^a ^b Sathasivam, Saratha. "Logic Learning in Hopfield Networks"  (pdf).
 3. [^] ^a ^b Storkey, Amos J., and Romain Valabregue. "The basins of attraction of a new Hopfield learning rule." *Neural Networks* 12.6 (1999): 869-876.
 4. [^] Hebb, Donald Olding. *The organization of behavior: A neuropsychological theory*. Lawrence Erlbaum, 2002.
 5. [^] Storkey, Amos. "Increasing the capacity of a Hopfield network without sacrificing functionality." *Artificial Neural Networks – ICANN97* (1997): 451-456.
 6. [^] ^a ^b Hertz, John A., Anders S. Krogh, and Richard G. Palmer. *Introduction to the theory of neural computation*. Vol. 1. Westview press, 1991.
 7. [^] Liou, C.-Y.; Lin, S.-L. (2006). "Finite memory loading in hairy neurons"  (PDF). *Natural Computing* 5 (1): 15–42. doi:10.1007/s11047-004-5490-x .
 8. [^] Liou, C.-Y.; Yuan, S.-K. (1999). "Error Tolerant Associative Memory"  (PDF). *Biological Cybernetics* 81: 331–342. doi:10.1007/s004220050566 .
- J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences of the USA*, vol. 79 no. 8 pp. 2554–2558, April 1982.
 - Hebb, D.O. (1949). *Organization of behavior*. New York: Wiley
 - Hertz, J., Krogh, A., & Palmer, R.G. (1991). *Introduction to the theory of neural computation*. Redwood City, CA: Addison-Wesley.
 - McCullough, W.S., & Pitts, W.H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-133
 - Polyn, S.M., & Kahana, M.J. (2008). Memory search and the neural representation of context. *Trends in Cognitive Sciences*, 12, 24-30.
 - Rizzuto, D.S., & Kahana, M.J. (2001). An autoassociative neural network model of paired-associate learning. *Neural Computation*, 13, 2075-2092.
 - Kruse, Borgelt, Klawonn, Moewes, Russ, Steinbrecher (2011). *Computational Intelligence*.

External links [[edit](#)]

- Chapter 13 [The Hopfield model](#)  of *Neural Networks - A Systematic Introduction*  by Raul Rojas (ISBN 978-3-540-60505-8)
- Hopfield Neural Network Applet 
- Hopfield Neural Network implementation in Ruby (AI4R) 
- The Travelling Salesman Problem  - Hopfield Neural Network JAVA Applet
- scholarpedia.org- Hopfield network  - Article on Hopfield Networks by John Hopfield
- Hopfield Network Learning Using Deterministic Latent Variables  - Tutorial by Tristan Fletcher
- Neural Lab Graphical Interface  - Hopfield Neural Network graphical interface (Python & gtk)



Wikimedia Commons has media related to [Hopfield net](#).

v · t · e	Stochastic processes	[hide]
Discrete time	Bernoulli process · Branching process · Chinese restaurant process · Galton–Watson process · Independent and identically distributed random variables · Markov chain · Moran process · Random walk (Loop-erased · Self-avoiding)	
Continuous time	Bessel process · Birth–death process · Brownian motion (Bridge · Excursion · Fractional · Geometric · Meander) · Cauchy process · Contact process · Continuous-time random walk · Cox process · Diffusion process · Empirical process · Feller process · Fleming–Viot process · Gamma process · Hunt process · Interacting particle systems · Itô diffusion · Itô process · Jump diffusion · Jump process · Lévy process · Local time · Markov additive process · McKean–Vasov process · Ornstein–Uhlenbeck process · Poisson process (Compound · Non-homogeneous · Point process) · Schramm–Loewner evolution · Semimartingale · Sigma-martingale · Stable process · Superprocess · Telegraph process · Variance gamma process · Wiener process · Wiener sausage	
Both	Branching process · Gaussian process · Hidden Markov model (HMM) · Markov process · Martingale (Differences · Local · Sub- · Super-) · Random dynamical system · Regenerative process · Renewal process · White noise	
Fields and other	Dirichlet process · Gaussian random field · Gibbs measure · Hopfield model · Ising model (Potts model · Boolean network) · Markov random field · Percolation · Pitman–Yor process · Point process (Cox · Poisson) · Random field · Random graph	
Time series models	Autoregressive conditional heteroskedasticity (ARCH) model · Autoregressive integrated moving average (ARIMA) model · Autoregressive (AR) model · Autoregressive–moving-average (ARMA) model · Generalized autoregressive conditional heteroskedasticity (GARCH) model · Moving-average (MA) model	
Financial models	Black–Derman–Toy · Black–Karasinski · Black–Scholes · Chen · Constant elasticity of variance (CEV) · Cox–Ingersoll–Ross (CIR) · Garman–Kohlhagen · Heath–Jarrow–Morton (HJM) · Heston · Ho–Lee · Hull–White · LIBOR market · Rendleman–Bartter · SABR volatility · Vašíček · Wilkie	
Actuarial models	Bühlmann · Cramér–Lundberg · Risk process · Sparre–Anderson	

Queueing models	Bulk · Fluid · Generalized queueing network · <i>M/G/1</i> · <i>MM/1</i> · <i>MM/c</i>
Properties	Càdlàg paths · Continuous · Continuous paths · Ergodic · Exchangeable · Feller-continuous · Gauss–Markov · Markov · Mixing · Piecewise deterministic · Predictable · Progressively measurable · Self-similar · Stationary · Time-reversible
Limit theorems	Central limit theorem · Donsker's theorem · Doob's martingale convergence theorems · Ergodic theorem · Fisher–Tippett–Gnedenko theorem · Large deviation principle · Law of large numbers (weak/strong) · Law of the iterated logarithm · Maximal ergodic theorem · Sanov's theorem
Inequalities	Burkholder–Davis–Gundy · Doob's martingale · Kunita–Watanabe
Tools	Cameron–Martin formula · Convergence of random variables · Doléans-Dade exponential · Doob decomposition theorem · Doob–Meyer decomposition theorem · Doob's optional stopping theorem · Dynkin's formula · Feynman–Kac formula · Filtration · Girsanov theorem · Infinitesimal generator · Itô integral · Itô's lemma · Kolmogorov continuity theorem · Kolmogorov extension theorem · Lévy–Prokhorov metric · Malliavin calculus · Martingale representation theorem · Optional stopping theorem · Prohorov theorem · Quadratic variation · Reflection principle · Skorokhod integral · Skorokhod's representation theorem · Skorokhod space · Snell envelope · Stochastic differential equation (Tanaka) · Stopping time · Stratonovich integral · Uniform integrability · Usual hypotheses · Wiener space (Classical · Abstract)
Disciplines	Actuarial mathematics · Econometrics · Ergodic theory · Extreme value theory (EVT) · Large deviations theory · Mathematical finance · Mathematical statistics · Probability theory · Queueing theory · Renewal theory · Ruin theory · Statistics · Stochastic analysis · Time series analysis · Machine learning
List of topics · Category	

Categories: Artificial neural networks

This page was last modified on 27 August 2015, at 04:32.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

