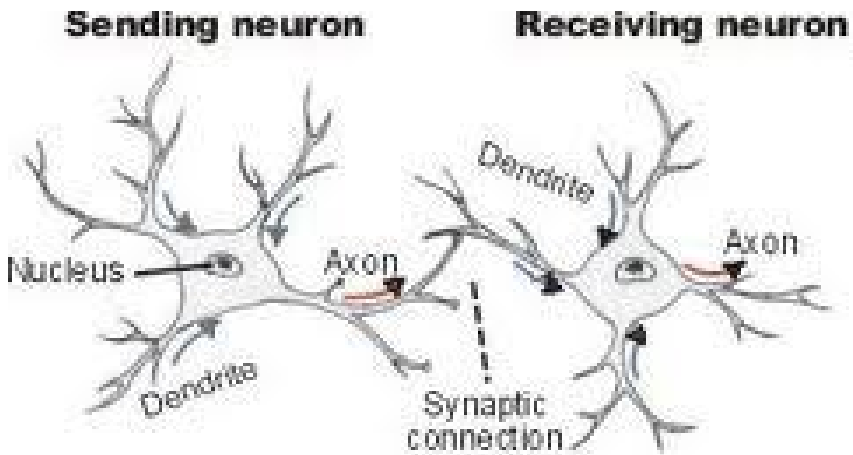
Neural Networks and their applications

The Hebbian rule in the brain

- Donald Hebb hypothesised in 1949 how neurons are connected with each other in the brain:
- “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.”

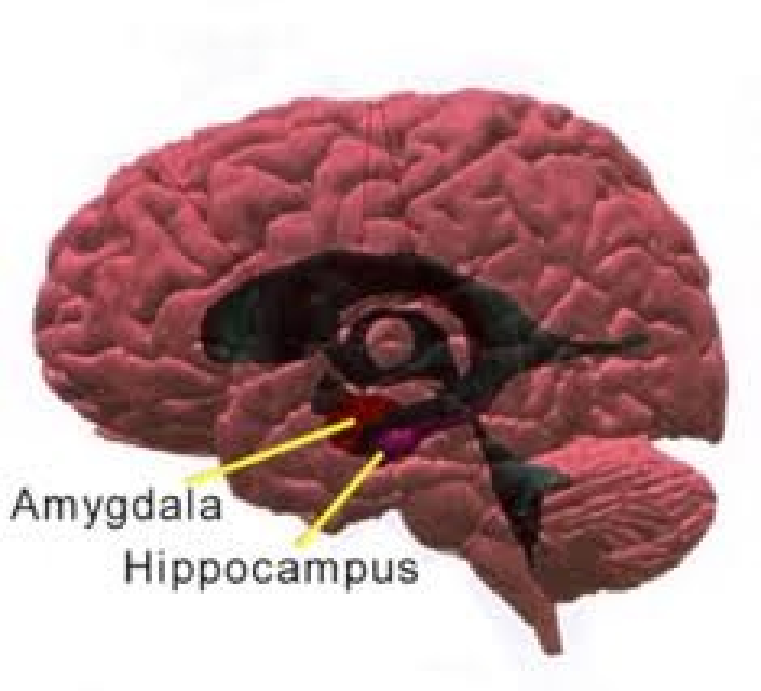


The Hebbian rule in the brain

- **Long Term Potentiation** (LTP) was established as a main paradigm in neuroscience, confirming Hebb's insight.
- The simple slogan to describe LTP is:
- "Neurons that fire together, wire together. Neurons that fire out of sync, fail to link."
- The neural network stores and retrieves associations, which are learned as synaptic connection.

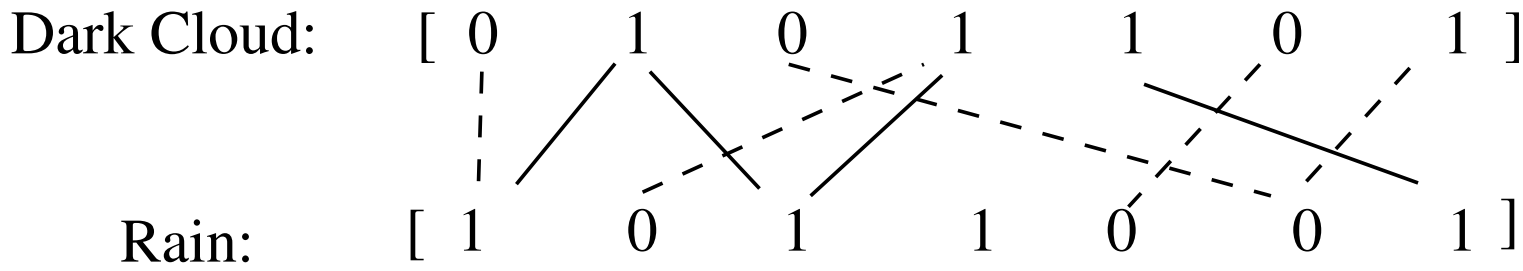
Human learning

- Learning is to associate two events with each other.
- The main brain organ for learning/explicit memory is the hippocampus (of the limbic system) using Hebbian type.



Explicit learning

- Consider two events “**Dark Cloud**” and “**Rain**”, represented for simplicity by two groups of 7 neurons below.
- Each is represented by the firing of particular neurons:



- Every (solid or dashed) line represents a synaptic connection from the terminal of a neuron in the first group to the dendrite of a neuron in the second.
- In Hebbian learning, synaptic modification only occurs between two firing neurons. In this case, these learning synaptic connections are given by the solid lines.

Human memory

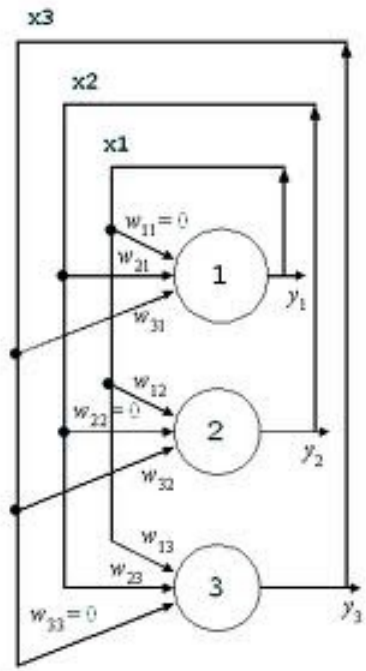
- Human memory thus works in an associative or content-addressable way.
- The memory of the individual is retrieved by a string of associations about the physical features, personality characteristics and social relations of that individual, which are dealt with by different parts of the brain.
- Human beings are also able to fully recall a memory by first remembering only particular aspects or features of that memory.

Unsupervised learning: The Hopfield network I

- In 1982, John Hopfield introduced an artificial neural network to store and retrieve memory like the human brain.
- Here, a neuron either is **on** (+1) or is **off** (-1), a vast simplification of the real situation.
- The state of a neuron (+1 or -1) will be renewed depending on the input it receives from other neurons.
- A Hopfield network is initially trained to store a number of patterns or memories.
- It is then able to recognise any of the learned patterns by exposure to only partial or even some corrupted information about that pattern, i.e., it eventually settles down and returns the closest pattern or the best guess.

The Hopfield network II

- A Hopfield network is single-layered and **recurrent** network: the neurons are fully connected, i.e., every neuron is connected to every other neuron.
- Given two neurons i and j there is a connectivity weight w_{ij} between them which is symmetric $w_{ij} = w_{ji}$ with zero self-connectivity $w_{ii} = 0$.
- Three neurons $i = 1, 2, 3$ with values ± 1, connectivity w_{ij}:



Updating rule

- Assume N neurons $= 1, \cdots, N$ with values $x_i = \pm 1$
- The update rule is:

$$\text{If } h_i \geq 0 \text{ then } 1 \leftarrow x_i \text{ otherwise } -1 \leftarrow x_i$$

  where $h_i = \sum_{j=1}^{N} w_{ij} x_j$.
- There are now two ways to update the nodes:
- **Asynchronously:** At each point in time, update one node chosen randomly or according to some rule.
- **Synchronously:** Every time, update all nodes together.
- Asynchronous updating, focused on here, is more biologically realistic.

A simple example

- Suppose we only have two neurons: $N = 2$.
- Then there are essentially two non-trivial choices for connectivities (i) $w_{12} = w_{21} = 1$ or (ii) $w_{12} = w_{21} = -1$.
- **Asynchronous updating:**
- In the case of (i) there are two attracting fixed points namely $[1, 1]$ and $[-1, -1]$. All orbits converge to one of these.
- For (ii), the attracting fixed points are $[-1, 1]$ and $[1, -1]$ and all orbits converge to one of these.

Energy function

- Hopfield networks have an energy function such that every time the network is updated asynchronously the energy level decreases (or is unchanged).
- For a given state (x_i) of the network and for any set of connection weights w_{ij} with $w_{ij} = w_{ji}$ and $w_{ii} = 0$, let

$$E = -\frac{1}{2} \sum_{i,j=1}^{N} w_{ij} x_i x_j$$

- We update x_m to x'_m and denote the new energy by E'.
- Then $E' \leq E$.
- The network eventually decreases to a stable equilibrium which is a local minimum of E.

Training the network: one pattern

- Training pattern $\vec{x} = (x_1, \ldots, x_i, \ldots, x_N) \in \{-1, 1\}^N$
- To construct a Hopfield network that remembers $\vec{x}$, we need to choose the connection weights w_{ij} appropriately.
- Choose $w_{ij} = \eta x_i x_j$ for $1 \leq i, j \leq N$ $(i \neq j)$, where $\eta > 0$ is the learning rate,
- Then the values x_i will not change under updating:
- We have

$$h_i = \sum_{j=1}^{N} w_{ij} x_j = \eta \sum_{j \neq i} x_i x_j x_j = \eta \sum_{j \neq i} x_i = \eta (N-1) x_i$$

- Thus the value of x_i, whether 1 or -1 will not change, so that $\vec{x}$ is a fixed point or an **attractor** of the network.

Neurons pull in or push away each other

- Consider the connection weight $w_{ij} = w_{ji}$ between two neurons i and j.
- If $w_{ij} > 0$, the updating rule implies:
  - when $x_j = 1$ then the contribution of j in the weighted sum, i.e. $w_{ij}x_j$, is positive. Thus x_i is pulled by j towards its value $x_j = 1$;
  - when $x_j = -1$ then $w_{ij}x_j$, is negative, and x_i is again pulled by j towards its value $x_j = -1$.
- Thus, if $w_{ij} > 0$, then i is pulled by j towards its value. By symmetry j is also pulled by i towards its value.
- It follows that for a given set of values $x_i \in \{-1, 1\}$ for $1 \leq i \leq N$, the choice of weights taken as $w_{ij} = x_i x_j$ for $1 \leq i \leq N$ corresponds to the Hebbian rule.

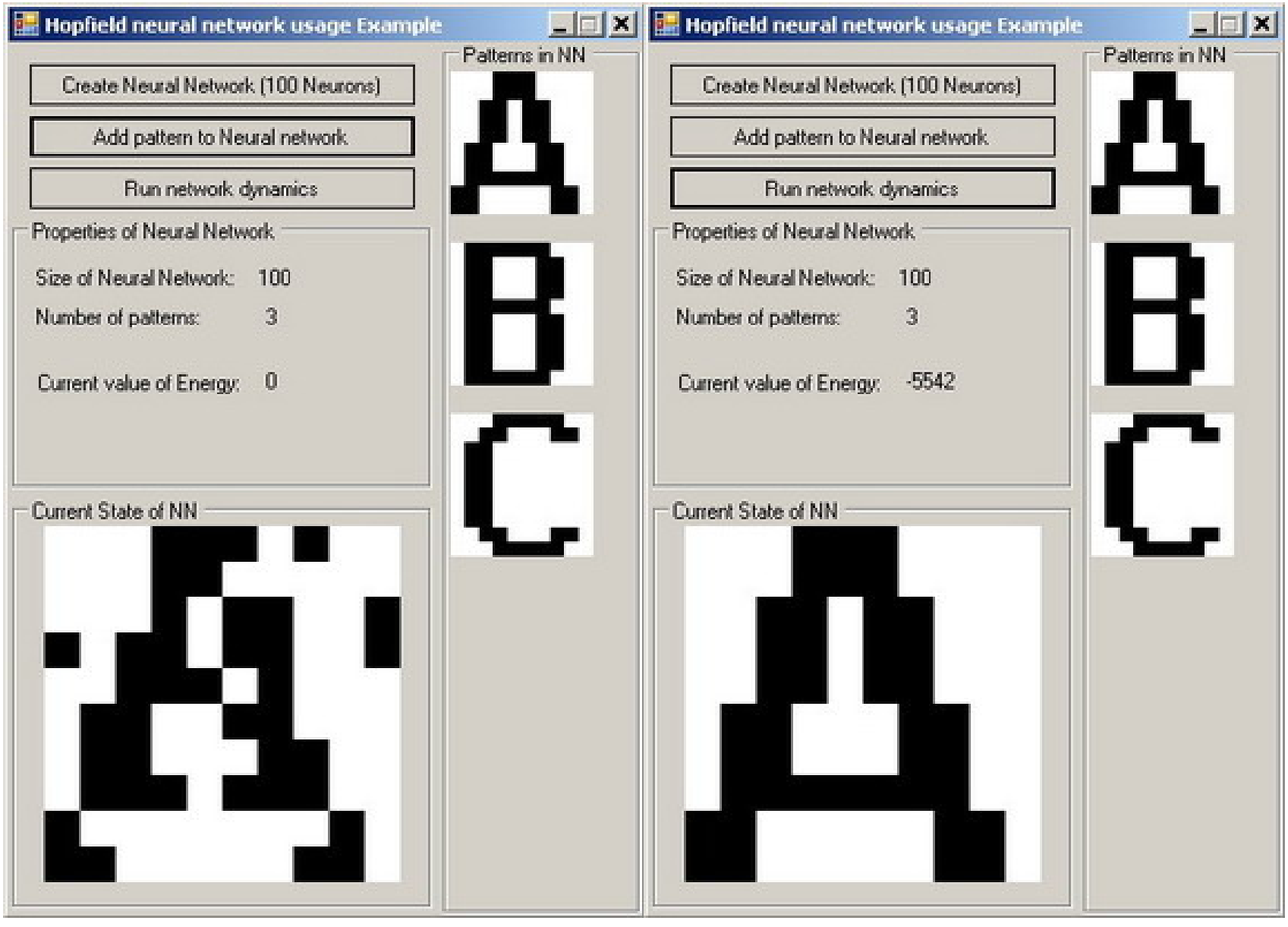
Training the network: Many patterns

- More generally, if we have p patterns $\vec{x}^k$, $k = 1, \ldots, p$, we choose $w_{ij} = \frac{1}{N} \sum_{k=1}^{p} x_i^k x_j^k$.
- If p/N is small then with high probability each pattern $\vec{x}^k$ becomes a fixed point of the network.

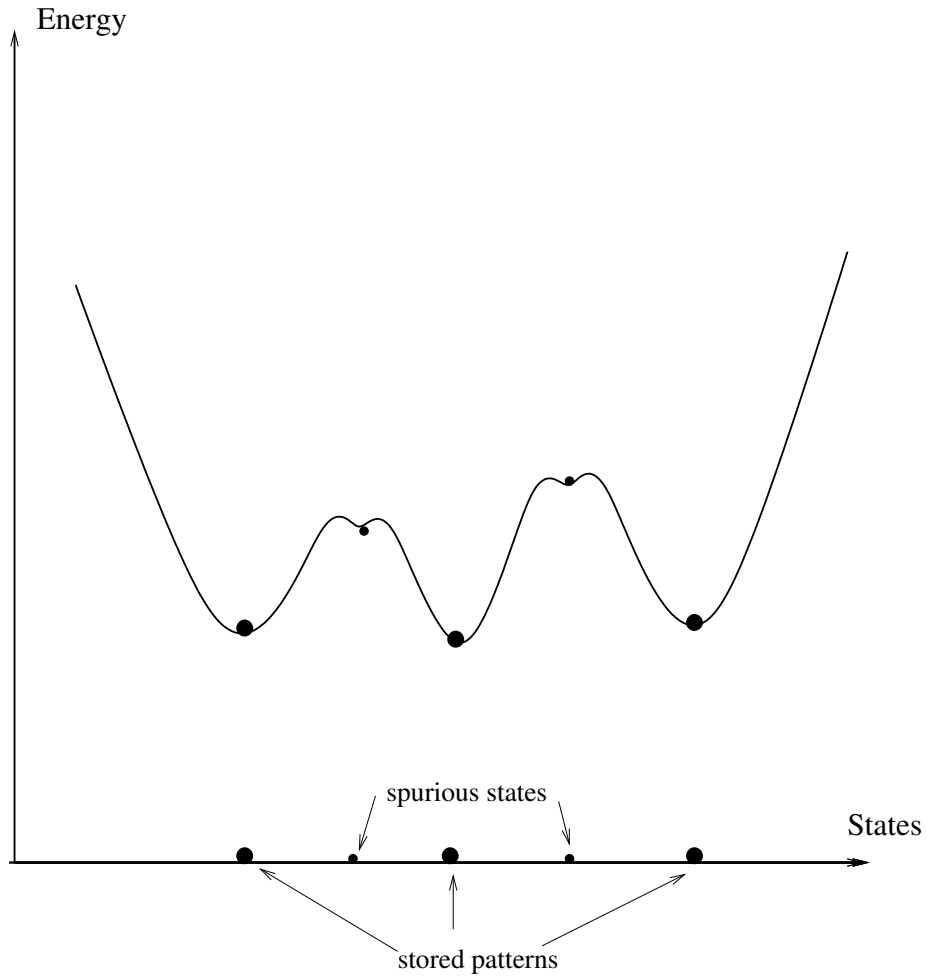
P_{error}	p/N
0.001	0.105
0.0036	0.138
0.01	0.185
0.05	0.37
0.1	0.61

- There are also some **spurious states** (fixed points of the network other than the p original patterns) for example $\pm\text{sgn}(\pm\vec{x}^{k_1} \pm \vec{x}^{k_2} \pm \vec{x}^{k_3})$ called a mixture state.

Pattern Recognition



Energy landscape



- Using a stochastic version of the Hopfield model one can eliminate or reduce the spurious states.

Some other neural networks

- **Boltzmann Machines** extend Hopfield networks to two layered networks. They can be used to recognise greyton images or probability distributions.
- **Feed-Forward Networks** such as the **perceptron** are hierarchically layered with non-symmetric, unidirectional, non-zero synaptic couplings only between neurons in each layer and those in the next layer.
- The values of input nodes (layer zero) and those of the output nodes (final layer) are specified as well as the rule for computing the values of the neurons at each layer in terms of those in the previous layer.
- The task is to obtain the synaptic couplings by some learning rule so that the final out put matches with the output nodes.
- These networks will not have an energy function like the Hopfield networks.