Hopfield Networks

The Hebbian rule

- 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."
- In experiments in 1966 and 1973, 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.
- In the Hebbian type of learning, both presynaptic and postsynaptic neurons are involved.
- The main brain organ for learning/explicit memory is the hippocampus (of the limbic system) using Hebbian type.

◆□▶ ◆□▶ ▲□▶ ▲□▶ ■ ののの



Explicit learning

- An event in the hippocampus is sculpted by a group of firing neurons.
- Consider two events "Dark Cloud" and "Rain", represented for simplicity by two groups of 7 neurons below.
- Dark Cloud is represented by the firing of neurons 2, 4, 5, 7 in the first group whereas Rain is represented by the firing of neurons 1, 3, 4, 7.



- 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.
- When a dark cloud and rain happen together, the two sets of neurons fire and the learning synapses are strengthened.

Human memory

- Human memory thus works in an associative or content-addressable way.
- There is no location in the neural network in the brain for a particular memory say of an individual.
- Rather, 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.
- Using advanced imaging technique, a sophisticated pattern of activation of various neural regions is observed in the process of recalling an individual.
- Human beings are also able to fully recall a memory by first remembering only particular aspects or features of that memory.

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 (firing) or is off (not firing), a vast simplification of the real situation.
- The state of a neuron (on: +1 or off: -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.
- Thus, like the human brain, the Hopfield model has stability in pattern recognition.
- With over 14,000 citations, Hopfield's original paper is the precursor of BM, RBM and DBN.

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.
- ▶ Below three neurons i = 1,2,3 with values x_i = ±1 have connectivity w_{ij}; any update has input x_i and output y_i.



Updating rule

- Assume *N* neurons = 1, \cdots , *N* with values $x_i = \pm 1$
- The update rule is for the node *i* is given by:

If $h_i \ge 0$ then $1 \leftarrow x_i$ otherwise $-1 \leftarrow x_i$

where $h_i = \sum_{j=1}^{N} w_{ij}x_j + b_i$ is called the **field** at *i*, with $b_i \in \mathbb{R}$ a bias.

- ► Thus, $x_i \leftarrow \text{sgn}(h_i)$, where sgn(r) = 1 if $r \ge 0$, and sgn(r) = -1 if r < 0.
- We put b_i = 0 for simplicity as it makes no difference to training the network with random patterns.
- We therefore assume $h_i = \sum_{j=1}^{N} w_{ij} x_j$.
- 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 is more biologically realistic.

Hopfield Network as a Dynamical system

- ► Take $X = \{-1, 1\}^N$ so that each state $x \in X$ is given by $x_i \in \{-1, 1\}$ for $1 \le i \le N$.
- > 2^N possible states or configurations of the network.
- Define a metric on X by using the Hamming distance between any two states:

$$H(\mathbf{x},\mathbf{y}) = \#\{i : \mathbf{x}_i \neq \mathbf{y}_i\}$$

- ► H is a metric with 0 ≤ H(x, y) ≤ N: it is clearly reflexive and symmetric, check the triangular inequality!
- With either the asynchronous or synchronous updating rule, we get a discrete time dynamical system:
- The updating rule Up : $X \rightarrow X$ defines a map.
- And Up : $X \rightarrow X$ is trivially continuous; check!
- Interested in the long term behaviour of orbits, as before.

A simple example

- Suppose we only have two neurons: N = 2.
- ► Then there are essentially two non-trivial choices for connectivities (i) w₁₂ = w₂₁ = 1 or (ii) w₁₂ = w₂₁ = -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. Therefore, in both cases, the network is **sign blind**: for any attracting fixed point, swapping all the signs gives another attracting fixed point.
- Synchronous updating: In both cases of (i) and (ii), although there are fixed points, none attract nearby points, i.e., they are not attracting fixed points. There are also orbits which oscillate forever.

Energy function

- Hopfield networks have an energy function which decreases or is unchanged with asynchronous updating.
- For a given state x ∈ {−1, 1}^N of the network and for any set of connection weights w_{ij} with w_{ij} = w_{ji} and w_{ii} = 0, let

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

- ▶ We update *x_m* to *x'_m* and denote the new energy by *E'*.
- **Exercise:** Show that $E' E = (x_m x'_m) \sum_{i \neq m} w_{mi} x_i$.
- Using the above equality, if $x_m = x'_m$ then we have E' = E.
- ▶ If $x_m = -1$ and $x'_m = 1$, then $x_m x'_m = -2$ and $h_m = \sum_i w_{mi} x_i \ge 0$. Thus, $E' E \le 0$.
- ▶ Similarly if $x_m = 1$ and $x'_m = -1$, then $x_m x'_m = 2$ and $h_m = \sum_i w_{mi} x_i < 0$. Thus, E' E < 0.

・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・

• Note: If x_m flips then $E' - E = 2x_m h_m$.

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.
- If w_{ij} < 0 however, then i is pushed away by j from its value and vice versa.
- ► It follows that for a given set of values $x_i \in \{-1, 1\}$ for 1 < i < N the choice of weights taken as $w_i = x x_i$ for
 - $1 \le i \le N$, the choice of weights taken as $w_{ij} = x_i x_j$ for
 - $1 \le i \le N$ corresponds to the Hebbian rule.

Training the network: one pattern ($b_i = 0$)

- Suppose the vector x̄ = (x₁,...,x_i,...,x_N) ∈ {−1,1}^N is a pattern we like to store in the memory of a Hopfield network.
- To construct a Hopfield network that remembers x, we need to choose the connection weights w_{ij} appropriately.
- If we choose w_{ij} = ηx_ix_j for 1 ≤ i, j ≤ N (i ≠ j), where η > 0 is the learning rate, then the values x_i will not change under updating as we show below.
- We have

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

- ► This implies that the value of x_i, whether 1 or −1 will not change, so that x̄ is a fixed point.
- ► Note that -x also becomes a fixed point when we train the network with x confirming that Hopfield networks are sign blind.

Training the network: Many patterns

- ► More generally, if we have p patterns \vec{x}^{ℓ} , $\ell = 1, ..., p$, we choose $w_{ij} = \frac{1}{N} \sum_{\ell=1}^{p} x_i^{\ell} x_j^{\ell}$.
- This is called the generalized Hebbian rule.
- We will have a fixed point x^k for each k iff sgn(h^k_i) = x^k_i for all 1 ≤ i ≤ N, where

$$h_{i}^{k} = \sum_{j=1}^{N} w_{ij} x_{j}^{k} = \frac{1}{N} \sum_{j=1}^{N} \sum_{\ell=1}^{p} x_{i}^{\ell} x_{j}^{\ell} x_{j}^{\ell} x_{j}^{\ell}$$

Split the above sum to the case $\ell = k$ and the rest:

$$h_i^k = x_i^k + \frac{1}{N} \sum_{j=1}^N \sum_{\ell \neq k} x_i^\ell x_j^\ell x_j^\ell x_j^k$$

- If the second term, called the crosstalk term, is less than one in absolute value for all *i*, then h_i^k will not change and pattern k will become a fixed point.
- In this situation every pattern x^k becomes a fixed point and we have an associative or content-addressable memory.

Pattern Recognition



Stability of the stored patterns

- How many random patterns can we store in a Hopfield network with N nodes?
- In other words, given N, what is an upper bound for p, the number of stored patterns, such that the crosstalk term remains small enough with high probability?
- Multiply the crosstalk term by $-x_i^k$ to define:

$$C_i^k := -x_i^k \frac{1}{N} \sum_{j=1}^N \sum_{\ell \neq k} x_i^\ell x_j^\ell x_j^k$$

- If C^k_i is negative, then the crosstalk term has the same sign as the desired x^k_i and thus this value will not change.
- ► If, however, C^k_i is positive and greater than 1, then the sign of h_i will change, i.e., x^k_i will change, which means that node i would become unstable.
- We will estimate the probability that $C_i^k > 1$.

Distribution of C_i^k

- ▶ For $1 \le i \le N$, $1 \le \ell \le p$ with both *N* and *p* large, consider x_i^{ℓ} as purely random with equal probabilities 1 and -1.
- ▶ Thus, C_i^k is 1/N times the sum of (roughly) Np independent and identically distributed (i.i.d.) random variables, say y_m for $1 \le m \le Np$, with equal probabilities of 1 and -1.
- Note that $\langle y_m \rangle = 0$ with variance $\langle y_m^2 \langle y_m \rangle^2 \rangle = 1$ for all *m*.
- **Central Limit Theorem:** If z_m is a sequence of i.i.d. random variables each with mean μ and variance σ^2 then for large *n*

$$X_n = \frac{1}{n} \sum_{m=1}^n z_m$$

has approximately a normal distribution with mean

 $\langle X_n \rangle = \mu$ and variance $\langle X_n^2 - \langle X_n \rangle^2 \rangle = \sigma^2/n$.

► Thus for large *N*, the random variable $p(\frac{1}{Np}\sum_{m=1}^{Np} y_m)$, i.e., C_i^k , has approximately a normal distribution $\mathcal{N}(0, \sigma^2)$ with mean 0 and variance $\sigma^2 = p^2(1/(Np)) = p/N$.

Storage capacity

Therefore if we store p patterns in a Hopfield network with a large number of N nodes, then the probability of error, i.e., the probability that C_i^k > 1, is:

$$P_{\text{error}} = P(C_i^k > 1) \approx \frac{1}{\sqrt{2\pi\sigma}} \int_1^\infty \exp(-x^2/2\sigma^2) \, dx$$
$$= \frac{1}{2} (1 - \operatorname{erf}(1/\sqrt{2\sigma^2})) = \frac{1}{2} (1 - \operatorname{erf}(\sqrt{N/2p}))$$

where the error function erf is given by:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-s^2) \, ds.$$

Therefore, given N and p we can find out the probability Perror of error for a single neuron of a stored pattern.

Storage capacity



- The table shows the error for some values of p/N.
- A long and sophisticated analysis of the stochastic Hopfield network shows that if p/N > 0.138, small errors can pile up in updating and the memory becomes useless.

• The storage capacity is $p/N \approx 0.138$.

Spurious states

- Therefore, for small enough p, the stored patterns become attractors of the dynamical system given by the synchronous updating rule.
- ► However, we also have other, so-called **spurious states**.
- Firstly, for each stored pattern \vec{x}^k , its negation $-\vec{x}^k$ is also an attractor.
- Secondly, any linear combination of an odd number of stored patterns give rise to the so-called mixture states, such as

$$ec{x}^{ ext{mix}} = \pm ext{sgn}(\pm ec{x}^{k_1} \pm ec{x}^{k_2} \pm ec{x}^{k_3})$$

- Thirdly, for large p, we get local minima that are not correlated to any linear combination of stored patterns.
- If we start at a state close to any of these spurious attractors then we will converge to them. However, they will have a small basin of attraction.

Energy landscape



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

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Strong Patterns

- So far we have been implicitly assuming that each pattern in the Hopfield network is learned just once.
- In fact, this assumption follows from the condition that the stored patterns are random.
- A strong pattern is one that is multiply learned in the network.
- The **degree** of a pattern is its multiplicity.
- Thus, strong patterns have degree greater than 1.
- A pattern with degree 1 is called a **simple** pattern.
- Strong patterns are strongly stable and have large basins of attraction compared to simple patterns.
- Strong patterns are used to model behavioural and cognitive prototypes.
- We now consider the behaviour of Hopfield networks when both simple and strong patterns are present.

Strong Patterns

Experiment with smiley faces with six basic emotions, i.e., angry, disgusted, fearful, happy, sad and surprised:



We will create brains that interpret events usually with a particular emotion, for example in a sad or a happy way.

- Store the sad face three times and up to 800 random patterns in a network with 650 nodes.
- Exposing the network with any random pattern will with retrieve the sad face with negligible error (wrt the Hamming distance).
- ▶ Note that $803/(650 \times 0.138) \approx 8.95 \approx 9 = 3^2$.
- Now, in addition, store the happy face five times in the network.
- Any random pattern now retrieves the happy face with negligible error.
- In fact, we can store a total of 2248 random patterns and still retrieve the happy face with negligible error.
- ▶ Note that $2248/(650 \times 0.138) \approx 25.06 \approx 25 = 5^2$
- ► It seems that a strong pattern of degree *d* can be retrieved in the presence of up to N × 0.138 × d².
- This means that the capacity of the Hopfield network to retrieve a strong pattern to increases by the square of the degree of the strong pattern.

A typical updating sequence for a network with 5 copies of the happy face, 3 copies of the sad face and 2200 random patterns:



Learning strong patterns

- Assume we have p₀ distinct patterns x^ℓ where 1 ≤ ℓ ≤ p₀ with degrees d_ℓ for 1 ≤ ℓ ≤ p₀ respectively.
- Assume ∑^p_{ℓ=1} d_ℓ = p, i.e., we still have a total of p patterns counting their multiplicity.
- The Hebbian rule now gives: $w_{ij} = \frac{1}{N} \sum_{\ell=1}^{p_0} d_\ell x_i^{\ell} x_j^{\ell}$.
- As before, we have a fixed point x^k for each k iff sgn(h^k_i) = x^k_i for all 1 ≤ i ≤ N, where

$$h_i^k = \sum_{j=1}^N w_{ij} x_j^k = rac{1}{N} \sum_{j=1}^N \sum_{\ell=1}^{p_0} d_\ell x_i^\ell x_j^\ell x_j^\ell$$

• Again split the above sum to the case $\ell = k$ and the rest:

$$h_i^k = d_k x_i^k + \frac{1}{N} \sum_{j=1}^N \sum_{\ell \neq k} d_\ell x_i^\ell x_j^\ell x_j^k$$

Now if the second term, the crosstalk term, is small relative to d_k for all *i*, then h^k_i will not change and pattern k will become a fixed point.

Stability of the strong patterns

- We proceed as in the case of simple patterns.
- Multiplying the crosstalk term by $-x_i^k$:

$$C_i^k := -x_i^k \frac{1}{N} \sum_{j=1}^N \sum_{\ell \neq k} d_\ell x_i^\ell x_j^\ell x_j^k$$

- If C_i^k is less than d_k, then the crosstalk term has the same sign as the desired x_i^k and thus this value will not change.
- If, however, C^k_i is greater than d_k, then the sign of h_i will change, i.e., x^k_i will change, which means that node i would become unstable.
- We will estimate the probability that $C_i^k > d_k$.

Distribution of C_i^k

- ► As in the case of simple patterns, for $j \neq i$ and $\ell \neq k$ the random variables $d_{\ell}x_{i}^{\ell}x_{i}^{\ell}x_{i}^{k}/N$ are independent.
- But now they are no longer identically distributed and we cannot invoke the Central Limit Theorem (CLT).
- However, we can use a theorem of Lyapunov's which gives a generalisation of the CLT when the random variables are independent but not identically distributed. This gives:

$$\mathcal{C}_i^k \sim \mathcal{N}\left(0, \sum_{\ell
eq k} d_\ell^2 / N
ight)$$

Assuming only x¹ is strong, we get C¹_i ∼ N(0, (p − d₁)/N). If in addition d₁ ≪ p, the probability of error in retrieving x¹ would be:

$$\mathsf{Pr}_{error} \approx \frac{1}{2} \left(1 - \mathsf{erf}(d_1 \sqrt{N/2p}) \right)$$

A square law of attraction for strong patterns

We can rewrite the error probability in this case as

$$\mathsf{Pr}_{error} pprox rac{1}{2} \left(1 - \mathsf{erf}(\sqrt{Nd_1^2/2p}) \right)$$

- The error probability is thus constant if d_1^2/p is fixed.
- For d₁ = 1, we are back to the standard Hopfield model where p = 0.138N is the capacity of the network.
- It follows that for d₁ > 1 with d₁ ≪ p the capacity of the network to retrieve the strong pattern x¹ is

 $p \approx 0.138 d_1^2 N$

which gives an increase in the capacity proportional to the square of the degree of the strong pattern.

- As we have seen this result is supported by computer simulations. It is also proved rigorously using the stochastic Hopfield networks.
- This justifies strong patterns to model behavioural/cognitive prototypes in psychology and psychotherapy.