Networks of Neural Computation

WK8 – Hopfield Networks

Contents

Contents

Hopfield Model

- A Hopfield Network is a model of associative memory. It is based on Hebbian learning but uses binary neurons.
- It provides a formal model which can be analysed for determining the *storage capacity* of the network.
- It is inspired in its formulation by statistical mechanics models (Ising model) for magnetic materials.
- It provides a path for generalising deterministic network models to the stochastic case.

Hopfield Model-1

Hopf. Model

- The associative memory problem is summarised as follows:

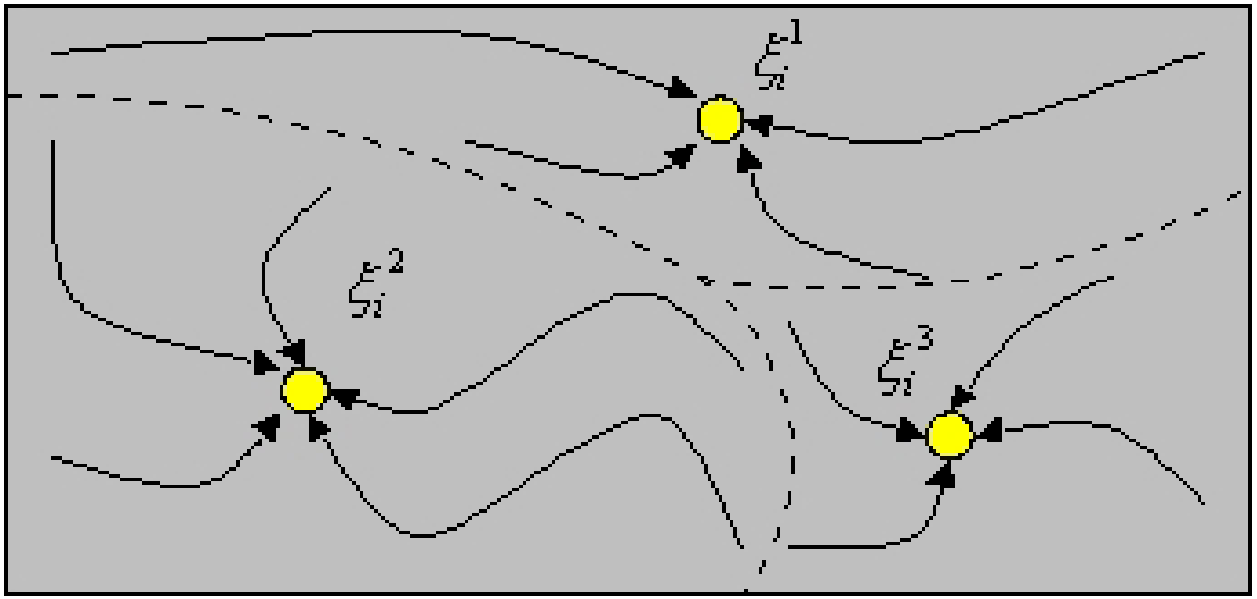
Store a set of p patterns ξ_i^μ in such a way that when presented with a new pattern ζ_i, the network responds by producing whichever one of the stored patterns most closely resembles ζ_i.

- The patterns are labelled μ=1,2,...,p, while the units in the network are labelled by i=1,2,...,N. Both the stored patterns, ξ_i^μ , and the test patterns, ζ_i , can be taken to be either 0 or 1 on a site i, though we will adopt a different convention henceforth.

- An associative memory can be thought as a set of **attractors**, each with its own **basin of attraction**.

Hopfield Model-2

- The dynamics of the system carries a starting points into one of the attractors as shown in the next figure.



Hopfield Model-3

Hopf. Model

- The Hopfiled model starts with the standard McCulloch-Pitts model of a neuron:

$$n_i(t+1) = \Theta\left(\sum_j w_{ij} n_j(t) - \mu_i\right)$$

Where Θ is the step function. In the Hopfield model the neurons have a binary output taking the values –1 and 1. Thus the model has the following form:

$$S_i = \operatorname{sgn}\left(\sum_j w_{ij} S_j - \vartheta_i\right)$$

Where the S_i and n_i are related thought the formula:

Hopfield Model-4

Hopf. Model

$S_i=2n_i-1$. The thresholds are also related by: $\theta_i=2\mu_i - \sum_j w_{ij}$, and the sgn(•) function is defined as:

$$\mathrm{sgn}(x)=\begin{cases} 1 & if \quad x \geq 0 \\ -1 & if \quad x < 0 \end{cases}$$

- For ease of analysis in what follows we will drop the thresholds ($\theta_i=0$) because we will analyse mainly random patterns and the thresholds are not very useful in this context. In this case the model is written as:

$$S_i = \mathrm{sgn}\left(\sum_j w_{ij} S_j\right)$$

Hopfield Model-5

- There are at least two ways in which we might carry out the updating specified by the above equation:
  - *Synchronously:* update all the units simultaneously at each time step;
  - *Asynchronously:* Update them one at a time. In this case we have two options:
    - At each time step, select at random a unit i to be updated, and apply the formula;
    - Let each unit independently choose to update itself according to the above formula, with some constant probability per unit time.

Hopfield Model-6

Hopf. Model

- We will study the *memorisation* (i.e. find a set of suitable w_{ij}) of a set of random patterns, which are made up of independent bits ξ_i which can each take on the values +1 and –1 with equal probability.
- Our procedure for testing whether a proposed form of w_{ij} is acceptable is first to see whether the patterns are themselves stable, and then to check whether small deviations from these patterns are corrected as the network evolves.
- We distinguish two cases:
  - One pattern
  - Many patterns

Hopfield Model-7: Storage of one pattern

- The condition for a single pattern to be stable is:

$$\xi_i = \mathrm{sgn}(\sum_j w_{ij}\xi_j) \qquad (for \quad all \quad i)$$

- It is easy to see that this is true if get the weight as proportional to the product of the components:

$$w_{ij} \propto \xi_i \xi_j$$

Since $\xi_i^2=1$. For convenience we get the constant of proportionality to be 1/N, where N is the number of units in the network. Thus we have:

$$w_{ij} = \frac{1}{N}\xi_i \xi_j$$

Hopfield Model-8: Storage of one pattern

- Furthermore, it is also obvious that even if a number (fewer than half) of the bits of the starting pattern S_i are wrong (i.e. not equal to ξ_i), they will be overwhelmed in the sum for the net input:

$$h_i = \sum_j w_{ij} S_j$$

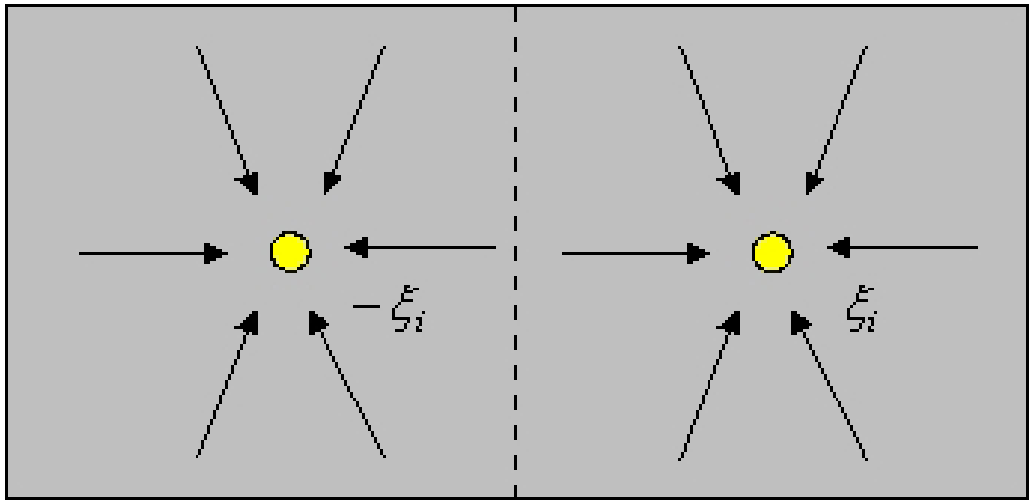
By the majority that are right and sgn(h_i) will still give ξ_i. This means that the network will correct errors as desired, and we can say that the pattern ξ_i is an **attractor**.

- Actually there are two attractors in this simple case; the other one is - ξ_i. This is called the **reversed state**. All starting configurations with

Hopfield Model-9: Storage of one pattern

Hopf. Model

more than half the bits different from the original pattern will end up in the reversed state. The configuration space is symmetrically divided into two basins of attraction, as shown in the next figure:



Hopfield Model-10: Storage of many patterns

- In the case of many patterns the weights are assumed to be a superposition of terms like in the case of a single pattern:

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \xi_i^{\mu}\xi_j^{\mu}$$

Where p is the number of patterns labeled by μ.

- Observe that this essentially the Hebb rule.

- An associative memory model using the Hebbian rule above for all possible pairs ij, with binary units and asynchronous updating is usually called a **Hopfield model**. The term also applies to variations.

Hopfield Model-11: Storage of many patterns

Hopf. Model

- Let us examine the stability of a particular pattern ξ_i^ν. The stability condition generalises to:

Where the net input h_i^ν to unit i in pattern ν is:

$$h_i^\nu \equiv \sum_j w_{ij}\xi_j^\nu = \frac{1}{N}\sum_j\sum_\mu \xi_i^\mu \xi_j^\mu \xi_j^\nu$$

Now we separate the sum on μ into the special term $\mu=\nu$ and all the rest:

$$h_i^\nu = \xi_i^\nu + \frac{1}{N}\sum_j\sum_{\mu\neq\nu} \xi_i^\mu \xi_j^\mu \xi_j^\nu$$

Hopfield Model-12: Storage of many patterns

- If the second term were zero, we could immediately conclude that pattern ν was stable according to the previous stability condition. This is still true if the second term is small enough: *if its magnitude is smaller that 1 it cannot change the sigh of* h_i^{ν} *and the stability condition will be still satisfied.*

- The second term is called **crosstalk**. It turns out that it is less than 1 in many cases of interest if p is small enough.

Hopfield Model-13: Storage Capacity

- Consider the quantity:

$$C_i^{\nu} = -\xi_i^{\nu} \frac{1}{N} \sum_j \sum_{\mu \neq \nu} \xi_i^{\mu} \xi_j^{\mu} \xi_j^{\nu}$$

If C_i^{ν} is negative the crosstalk term has the same sign as the desired ξ_i^{ν} and does no harm. But if its is positive and larger than 1, it changes the sign of h_i^{ν} and makes the bit i of pattern ν unstable.

- The C_i^{ν} depends on the patterns we try to store. For *random* patterns and with equal probability for the values +1 and –1 we can estimate the probability P_{error} that any chosen bit is unstable:

$$P_{error} = \text{Prob}(C_i^{\nu} > 1)$$

Hopfield Model-14: Storage Capacity

- Clearly P_{error} increases as we increase the number p of patterns. Choosing a criterion for acceptable performance (e.g. P_{error} <0.01) we can try to determine the **storage capacity** of the network: the maximum number of patterns that can stored without unacceptable errors.

- To calculate P_{error} we observe that C_i^{ν} behaves like a **binomial distribution** with zero mean and variance $\sigma^2 = p/N$, where p and N are assumed much larger than 1. For large values of N*p, we can approximate this distribution with a Gaussian distribution of the same mean and variance:

Hopfield Model-15: Storage Capacity

Hopf. Model

$$P_{error} = \frac{1}{\sqrt{2\pi\sigma}} \int_{1}^{\infty} e^{-x^2/2\sigma^2} dx$$

$$= \frac{1}{2}\left[1 - erf\left(\frac{1}{\sqrt{2\sigma^2}}\right)\right] = \frac{1}{2}\left[1 - erf\left(\sqrt{\frac{N}{2p}}\right)\right]$$

Where the **error function** erf(x) is defined by:

$$erf\ (x) \equiv \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-u^2} du$$

- The next table shows values of p/N required to obtain various values for P_{error}:

Hopfield Model-16: Storage Capacity

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

- This calculation tells us only about the *initial* stability of the patterns. If we choose p <0.185N, it tells us that no more than 1% of the pattern bits will be unstable initially.

- But if start the system in a particular pattern ξ_i^ν and about 1% of the bits flip, what happens next? It may be that the first few flips will cause more bits to

flip. In the worst case we will have an avalanche phenomenon. So, our estimates of p_{max} are really upper bounds. We may need smaller values of p to keep the final attractors close to the desired patterns.

- In summary, the capacity p_{max} is proportional to N (but never higher than 0.138N) if we are willing to accept a small percentage of errors in each pattern. It is proportional to N / log(N) if we insist that most of the patterns be recalled perfectly (this calculation will not be discussed).

Hopfield Model-18: Energy Function

Hopf. Model

- One of the most important contributions of Hopfiled was the introduction of an *energy function* into neural network theory. For the networks we consider this is:

$$H = -\frac{1}{2}\sum_{ij} w_{ij} S_i S_j$$

The double summation is over all i and j. The terms i=j are of no consequence because $S_i^2=1$; they just contribute a constant to H.

- The energy function is a function of the configuration $\{S_i\}$ of the system. We can imagine an **energy landscape** "above" the configuration space.

Hopfield Model-19: Energy Function

- The main property of an energy function is that *it always decreases (or remains constant) as the system evolves according to its dynamical rule.*
- Thus the attractors are the local minima of the energy surface.
- The concept of the energy function is very general and has many names in different fields: **Lyapunov function**, **Hamiltonian**, **Cost function**, **Objective function** and **Fitness function**.
- An energy function exists if the weights are symmetric, i.e. $w_{ij} = w_{ji}$. However the symmetry does not hold in general for neural networks.

Hopfield Model-20: Energy Function

Hopf. Model

- For symmetric weights we can write the energy function as follows:

$$H = C - \sum_{(ij)} w_{ij} S_i S_j$$

Where (ij) means all the distinct pairs ij, counting for example 12 as the same pair as 21. We exclude the ii terms. They give the constant C.

- It is now easy to see that the energy function will decrease under the dynamics of the Hopfiled model. Let S_i' be the new value of S_i for some unit i:

$$S_i' = \text{sgn}\left(\sum_j w_{ij} S_j \right)$$

Hopfield Model-21: Energy Function

- Obviously if $S_i' = S_i$ the energy is unchanged. In the other case $S_i' = - S_i$ so, picking out the terms that involves S_i:

$$H' - H = -\sum_{j \neq i} w_{ij} S_i' S_j + \sum_{j \neq i} w_{ij} S_i S_j$$

$$= 2 S_i \sum_{j \neq i} w_{ij} S_j$$

$$= 2 S_i \sum_{j} w_{ij} S_j - 2 w_{ii}$$

- The first term is negative from our previous hypothesis and the second term is term is negative because the Hebb rule gives $w_{ii} = p/N$ for all i. Thus

the energy decreases as it was claimed.

- The **self-coupling terms** w_{ii} may be omitted as the do not make any appreciable difference to the stability of the ξ_i^ν patterns in the large N limit.

- But they affect the dynamics and the number of the **spurious states** and it turns out that it is better to omit them. We can see easily why by simply separating the self-coupling term out of the dynamical rule:

$$S_i = \mathrm{sgn}\left(w_{ii} S_i + \sum_{j \neq i} w_{ij} S_j \right)$$

- If w_{ii} were larger than the sum of the other terms in some state, then $S_i = +1$ and $S_i = -1$ could *both* be

Hopfield Model-23: Energy Function

Hopf. Model

stable.

- This can produce additional stable spurious states in the neighbourhood of a desired attractor, reducing the size of the basin of attraction. If $w_{ii}=0$ then this problem does not arise; for a given configuration of the other S_i 's will always pick one of its states over the other.

Hopfield Model-24: Spurious States

- We have shown that the Hebb rule gives us a dynamical system which has attractors (the minima of the energy function). These are the desired patterns which have been stored and are called **retrieval states**.
- However the Hopfield model has other attractors as well. These are:
  - The *reversed states;*
  - The *mixture states*;
  - The *spin glass states*.

Hopfield Model-25: Spurious States

- The **reversed states** have been mentioned above and they are the result of the perfect symmetry in the dynamics of the Hopfield model between them and the desired patterns. We can eliminate them by following any agreed *convention*: For example we can reverse all the bits of a pattern if a specific bit has value –1.

- The **mixture states** are stable states which are not equal to any single pattern but instead correspond a linear combinations of an odd number of patterns. The simplest is a combination of three states:

Hopfield Model-26: Spurious States

$$\xi_i^{mix} = \mathrm{sgn}(\pm \xi_i^{\mu_1} \pm \xi_i^{\mu_2} \pm \xi_i^{\mu_3})$$

- The system does not choose an even number because the sum can be potentially zero, but the activation is allowed only to take values –1 / 1.
- There are also, for large p, local minima that are not correlated with any finite number of the original patterns ξ_i^μ. These are sometimes called **spin glass states** because of close correspondence to spin glass models in statistical mechanics.
- So the memory does not work perfectly; there are all these additional minima in addition to the ones we want. The second and the third classes are

Hopfield Model-27: Spurious States

called generally **spurious minima**.

Hopf. Model

- These have in general smaller basin of attraction than the retrieval states. We can use a number of ‘tricks’ such as *finite temperature* and biased patterns in order to reduce or remove them.

Magnetic Materials

- There is a close analogy between Hopfield networks and some simple models of **magnetic materials**. The analogy becomes particularly useful when we generalise the networks to use **stochastic units**, which brings the idea of **temperature** in network theory.
- A simple description of a magnetic material consists of a set of atomic magnets arranged on a regular lattice that represents the crystal structure of the material. We call the atomic magnets **spins**. In the simplest case the spins can have only two possible orientations: “up” (1) and “down” (-1).
- In a magnetic material each of the spins is influenced

the **magnetic field** h at its location. This magnetic field consists of any **external field** h^{ext} plus an **internal field** produced by the other spins. The contribution of each atom to the internal field at a given location is proportional to its own spin.

- Thus we have a magnetic field for location i:

$$h_i = \sum_j w_{ij} S_j + h^{ext}$$

- The coefficients w_{ij} measure the strength of influence of spin S_j on the field at S_i and are called **exchange interaction strengths**. It is always true for a magnet that $w_{ij} = w_{ji}$, i.e. the interactions are symmetric. They could be positive or negative.

Magnetic Materials-2

Magn. Mater.

- At *low temperature*, a spin tends to line up parallel to the local field h_i acting on it, so as to make $S_i=\mathrm{sgn}(h_i)$. This can happen asynchronously and in random order.
- Another way of specifying the interactions of the spins is be defining a potential energy function:

$$H = -\frac{1}{2}\sum_{ij} w_{ij} S_i S_j - h^{ext} \sum_i S_i$$

- Thus the match with the Hopfield model is complete:
  - Network weights → Exchange interaction strengths of the magnet;
  - Net input of neuron → Field acting on a spin (external field represents a threshold);
  - Network Energy function → Energy of magnet ($h^{ext}=0$)

Magnetic Materials-3

- McCulloach-Pitts rule → Dynamics of spins aligning with their local field.
- If the temperature is *not very low*, there is a complication to the magnetic problem. **Thermal fluctuations** tend to flip the spins, and thus upset the tendency of each spin to align with its field.
- The two influences, thermal fluctuations and field, are always present.Their relative strength depends on the temperature. In high temperatures the fluctuations dominate, while in lower ones the field dominates. In high temperatures is equally probable to find a spin in both “up” and “down” orientations.
- Keep in mind that there is not an equivalent idea of

"temperature" in the Hopfield model.

- The conventional way to describe mathematically the effect of thermal fluctuations in an Ising model is with the **Glauber dynamics**. We replace the previous deterministic dynamics by a *stochastic* rule:

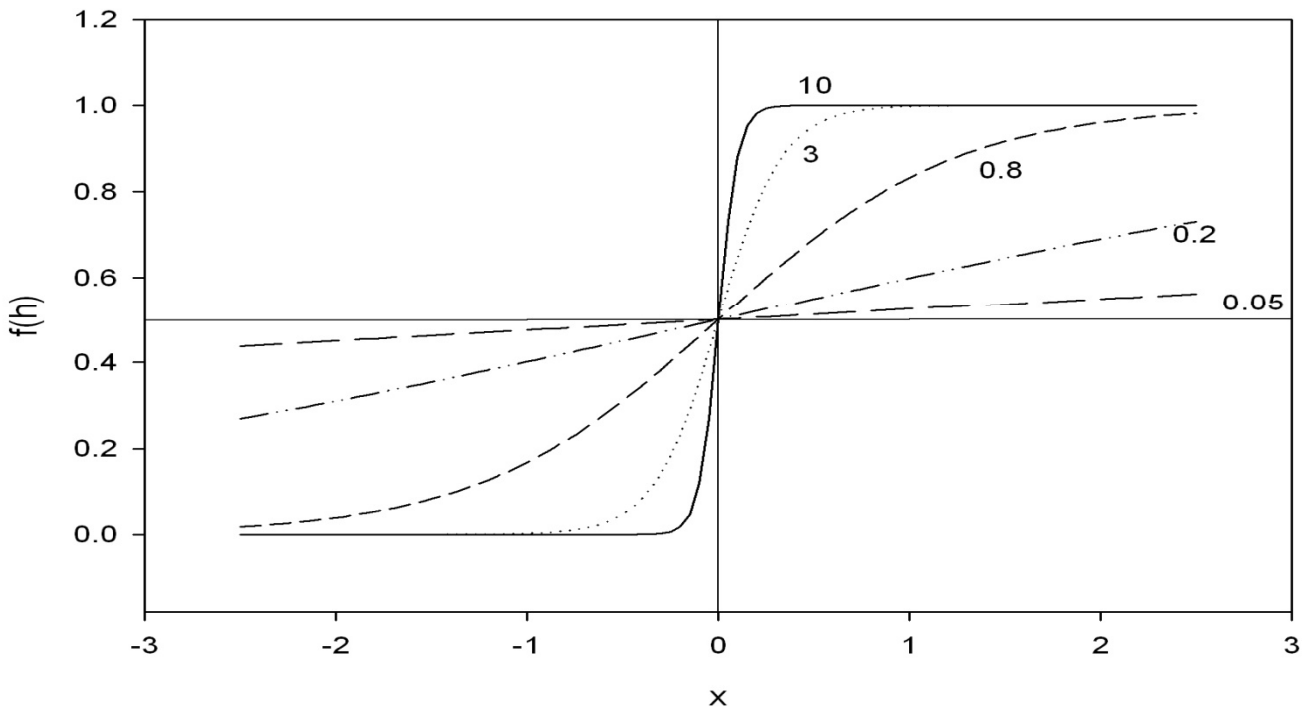
$$S_i = \begin{cases} +1 \quad with \quad probabilit\ y \quad g(h_i); \\ -1 \quad with \quad probabilit\ y \quad 1 - g(h_i) \end{cases}$$

- This is taken to be applied whenever the spin S_i is updated. The function g(h) depends on the temperature. There are several choices. The usual "Glauber" choice is a sigmoid-shaped function:

$$g(h) = f_\beta(h) \equiv \frac{1}{1+\exp(-2\beta h)}$$

Magnetic Materials-5

- A graph of the function is shown in the next figure for various values of the parameter β.



- β is related to the *absolute temperature T* by:

$$\beta = \frac{1}{k_B T}$$

Where k_B is the **Boltzmann's constant**.

- Because $1-f_\beta(h)=f_\beta(-h)$ we can write the probability in a symmetrical form:

$$\Pr ob(S_i = \pm 1) = f_\beta(\pm h_i) = \frac{1}{1+\exp(\mp 2\beta h_i)}$$

Magnetic Materials-7: Case of single spin

- We apply the Glauber dynamics to the case of a single spin in a fixed external field. With only one spin we can drop the subscripts.
- We can calculate the **average magnetisation** <S> by:

$$\langle S \rangle \equiv \Pr ob(+1) * (+1) + \Pr ob(-1) * (-1)$$

$$= \frac{1}{1 + \exp(-2\beta h)} - \frac{1}{1 + \exp(2\beta h)} = \tanh(\beta h)$$

Where tanh(•) is the **hyperbolic tangent** function.

- This result also applies to a whole collection of N spins if they experience the same external field and have no influence on one another. Such a system is called **paramagnetic**.

- When there are many interacting spins the problem is not solved easily. The evolution of spin S_i depends on h_i which itself involves other spins S_j which fluctuate randomly back and forth.
- There is no general way to solve the N spin problem *exactly* but there is an *approximation* which is sometimes quite good. It is known as **mean field theory** and consists of replacing the true fluctuating h_i by its average value:

$$\langle h_i \rangle = \sum_j w_{ij} \langle S_j \rangle + h^{ext}$$

- We can then compute the average $< S_j >$ just as in the single spin case:

Magnetic Materials-9: Mean Field Theory

$$\langle S_i \rangle = \tanh(\beta \langle h_i \rangle) = \tanh\left(\beta \sum_j w_{ij} \langle S_j \rangle + \beta h^{ext} \right)$$

- These are N nonlinear equations in N unknows but at least the do not involve stochastic variables.
- This mean field approximation often becomes exact in the limit of **infinite range interactions**, where each spin interacts with all the others. This happens because then the h_j is the sum of very many terms, and a central limit theorem can be applied.
- Even for short range interactions, where $w_{ij} \approx 0$ if spins i and j are more than a few lattice sites apart, the approximation can give a good qualitative description of the phenomena.

- In a **ferromagnet** all the w_{ij}'s are positive. Thus the spins tend to line up with each other, while thermal fluctuations tend to disrupt this ordering.
- There is a critical temperature T_c above of which thermal fluctuations win, making $\langle S \rangle = 0$, while beneath this the spin interactions win with $\langle S \rangle \neq 0$, which is the same in every site. In other words the system exhibits **phase transitions** at T_c.
- The simplest model of a ferromagnet is one in which all the weights are the same:

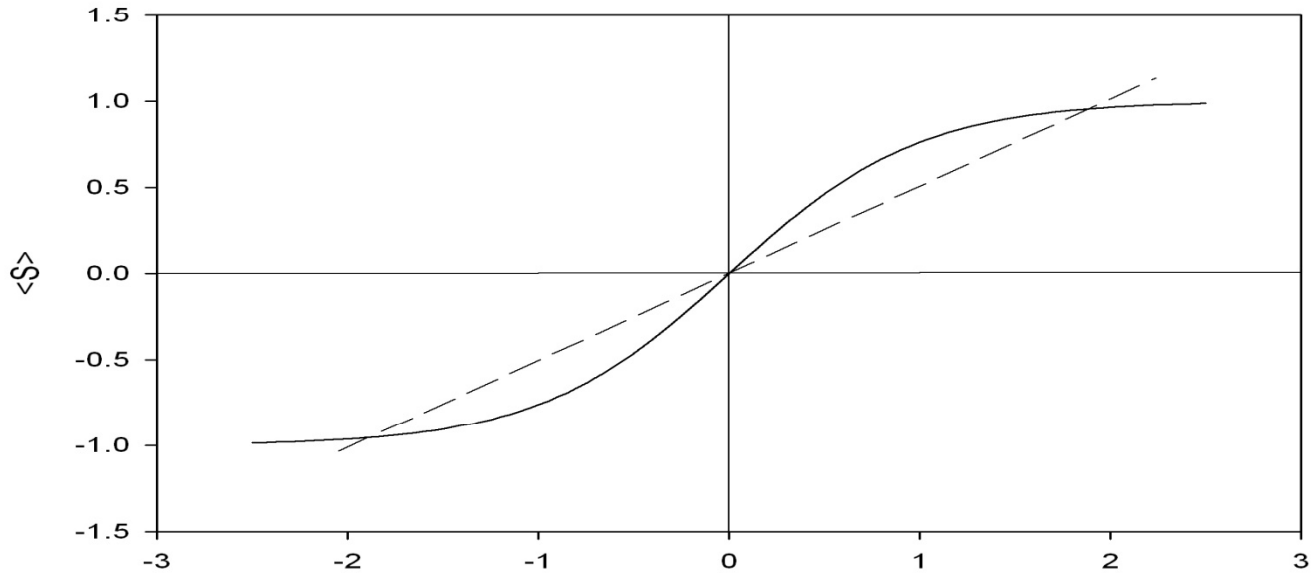
$$w_{ij} = \frac{J}{N} \quad (for \;\; all \;\; ij)$$

- J is a constant and N is the number of spins.
- For *zero temperature* this infinite range ferromagnet corresponds precisely (for J=1) to the one pattern Hopfield model for a pattern with ξ_i=1 for all i.
- At finite temperature we can use the mean field theory. In a ferrogmanetic state the magnetisation is uniform, i.e. $\langle S_i \rangle = \langle S \rangle$. Thus we can calculate $\langle S \rangle$ by simply solving the equation:

$$\langle S \rangle = \tanh(\beta J \langle S \rangle)$$

- Here we have set h^{ext}=0 for convenience, but the generalisation is obvious.
- We can solve graphically the above equation as a

function of T:



- The type of solutions depend on whether βJ is smaller or larger than 1. This corresponds to the different behaviour above and below the critical

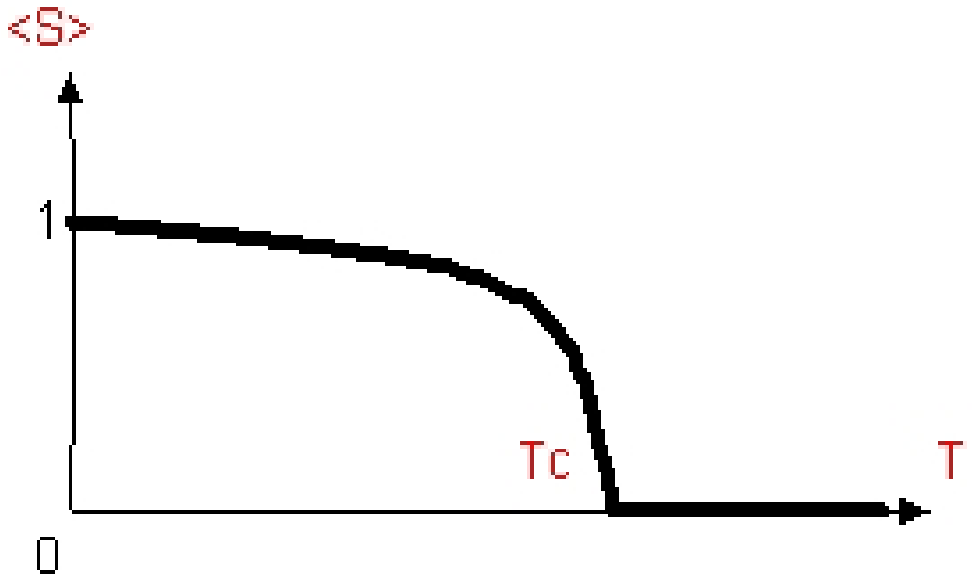
temperature:

- When $T \geq T_c$ there is only the trivial solution $\langle S \rangle = 0$;
- When $T < T_c$ there are two other solutions with $\langle S \rangle \neq 0$, one the negative of the other. Both are stable with the solution $\langle S \rangle = 0$ is unstable.

- The magnitude of the **average magnetisation** $\langle S \rangle$ rises sharply (continuously, but with infinite derivative at $T=T_c$) as one goes below T_c. As T approaches 0, $\langle S \rangle$ approaches ± 1; all spins point in the same direction. See next figure:

Magnetic Materials-14: Mean Field Theory

Magn. Mater.



Stochastic Networks

- We now apply the previous results to neural networks, making the units stochastic, applying the mean field theory and calculating eventually the storage capacity.

Stoch. Nets

- We can make our units stochastic by using the same rule as for the spins of the Ising model, i.e.:

$$\Pr ob(S_i = \pm 1) = f_\beta(\pm h_i) = \frac{1}{1 + \exp(\mp 2\beta h_i)}$$

- We use the above rule for neuron S_i whenever is selected for updating and select units in random order as before. The function $f_\beta(h)$ is called **logistic function**.

Stochastic Networks-1

- What is the meaning of this stochastic bahaviour? It actually captures a number of facts on real neurons:
  - Neurons fire with variable strength;
  - Delays in responses;
  - Random fluctuations from release of transmitters in discrete vesicles;
  - Other factors.
- These effects can be thought as **noise** and can be represented by the thermal fluctuations as in the case of the magnetic materials. Parameter β is not involved with any real temperature. Simply controls the noise level.

Stochastic Networks-2

- However, it is useful to define a pseudo-temperature T for the network by:

$$\beta \equiv \frac{1}{T}$$

- The temperature T controls the steepness of the sigmoid $f_\beta(h)$ near h=0. At very low temperature the sigmoid becomes the step function and the stochastic rule reduces to the deterministic McCulloch-Pitts rule for the original Hopfield network. As T increases this sharp threshold is softened up in a stochastic way.

- The use of a stochastic unit is not only for mathematical convenience, but also because it makes possible to kick the system out of spurious local minima of the energy function. The spurious states,

will be in general less stable (higher in energy) than the retrieval patterns and they will not trap a stochastic system permanently.

- Because the system is stochastic it will involve in a different way every time that it runs. Thus the only meaningful quantities to calculate are averages, weighted by the probabilities of each history.

- However, to apply the statistical mechanics methods we need the system to come to *equilibrium*. This means that averge quantities such as $<S_i>$ become eventually *time-independent*. Networks with an energy function do come to equilibrium.

- We can now apply the mean field approximation to the stochastic model which we have defined and we will use the Hebb rule for the weights.

- We restrict ourselves to the case of p << N. Technically the analysis here is correct for any fixed p as N → ∞.

- By direct analogy to the case of the magnetic materials we can write:

$$\langle S_i \rangle = \tanh\left(\frac{\beta}{N} \sum_{j,\mu} \xi_i^{\mu} \xi_j^{\mu} \langle S_j \rangle \right)$$

- These equations are not solvable since they have N unknowns with N nonlinear equations. But we can make a hypothesis taking $\langle S_i \rangle$ proportional to one

of the stored patterns:

$$\langle S_i \rangle = m \, \xi_i^{\nu}$$

- We have seen that states are stable in the deterministic limit so we look for similar average states in the stochastic case.

- We have by application of the hypothesis to mean field equation above:

$$m \, \xi_i^{\nu} = \tanh\left(\frac{\beta}{N} \sum_{j,\mu} \xi_i^{\mu} \xi_j^{\mu} m \, \xi_j^{\nu} \right)$$

- Just as in the case of the deterministic network, the argument in the sigmoid can be split into a term proportional to ξ_i^{ν} and a cross talk term. In the limit of

p << N the crosstalk term is negligible and we have:

$$m\,\xi_i^{\nu} = \tanh(\beta m\,\xi_i^{\nu}) \Rightarrow m = \tanh(\beta m)$$

- This equation is of the same as in the case of the ferromagnet. It can be solved in the same graphical way. The memory states will be stable for temperatures than 1. Thus the critical temperature T_c will be 1 for the stochastic network in case p<<N.
- The number m by be written as:

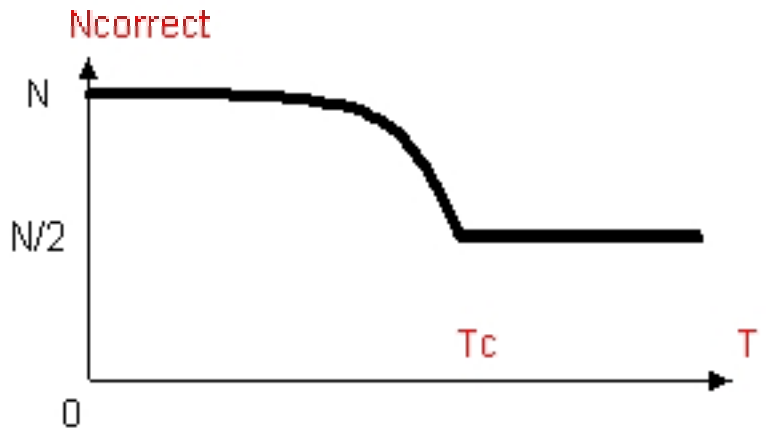
$m = \langle S_i \rangle / \xi_i^{\nu}$ =Prob(bit i is correct) – prob(bit i is incorrect)

- And thus the average number of correct bits in the

retrieved pattern is:

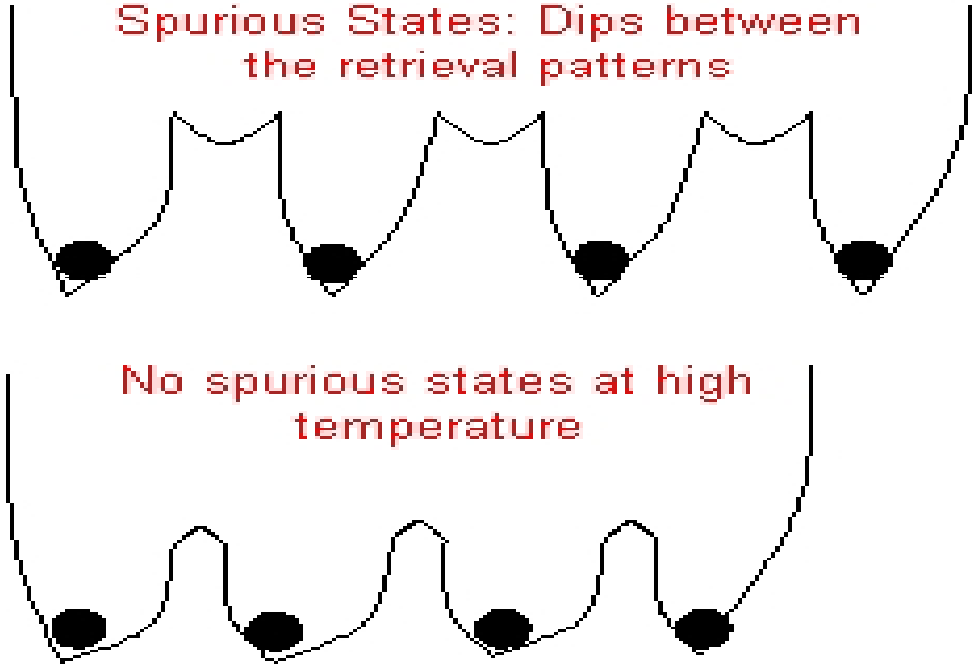
$$\langle N_{correct} \rangle = \frac{1}{2} N(1+m)$$

- This is shown in the next figure. Note that above the critical temperature the expected number is N/2 (as it is expected for random patterns), while at low temperature <$N_{correct}$> goes to N.



- The sharp change in behaviour at a particular noise level is another example of **phase transition**. One might assume that the change will be smooth, but this is not so in many cases in large systems.
- This means that the network ceases to function at all if a certain noise level is exceeded.
- The system is not a perfect device, even at low temperatures. There are still spurious states. The spin glass states are not relevant for p<<N but the reversed and the mixture states are both present.
- However, each type of mixture state has its own critical temperature, above which it is no longer stable.

- The next figure shows this schematically:



- The *highest* of the critical temperatures is 0.46, for the combinations of three patterns. So, for $0.46<T<1$

there are no mixture states and only the desired patterns remain. This shows that noise can be useful for improving the performance of the network.

- To calculate the capacity of the network in the case where p is *of the order* of N we need to derive the mean field equations for this limit. However, we will not do this calculation but we will rather present the results. First we need to define some useful variables:

  - The **load parameter** is defined as:

$$\alpha = \frac{p}{N}$$

i.e. the number of patterns we try to store as a fraction of the number of units in the network. Now it is of order O(1), while in the previous analysis it was of order

O(1/N). We can freely use the N → ∞ limit in order to drop lower order terms;

- In this case, p ~ N, and we cannot drop the crosstalk term in the mean field equations, as we have done before. Now we have to pay attention to the overlaps of the state $\langle S_i \rangle$ and the patterns:

$$m_\nu = \frac{1}{N} \sum_i \xi_i^\nu \langle S_i \rangle$$

for all patterns, not just the one being retrieved. We suppose that it is pattern number 1 which we are interested in. Then m_1 is of order O(1) while each m_ν for $\nu \neq 1$ is small and of order $O(1/\sqrt{N})$ for our random patterns. Nevertheless the quantity:

$$r = \frac{1}{\alpha} \sum_{\nu \neq 1} m_\nu^2$$

which is the mean square overlap of the system configuration with the nonretrieved patterns, is of order unity. The factor $1/\alpha=N/p$ makes r a true overage over the (p-1) squared overlaps and cancels the expected $1/\sqrt{N}$ dependence of the m_ν's.

- It can be provided that the mean filed equations lead to the following system of self-consistent variables:

$$C = \sqrt{\frac{2}{\pi\alpha r}} \exp\left(-\frac{m^2}{2\alpha r}\right)$$

$$r = \frac{1}{(1 - C)^2}$$

$$m = erf\left(\frac{m}{\sqrt{2\alpha r}}\right)$$

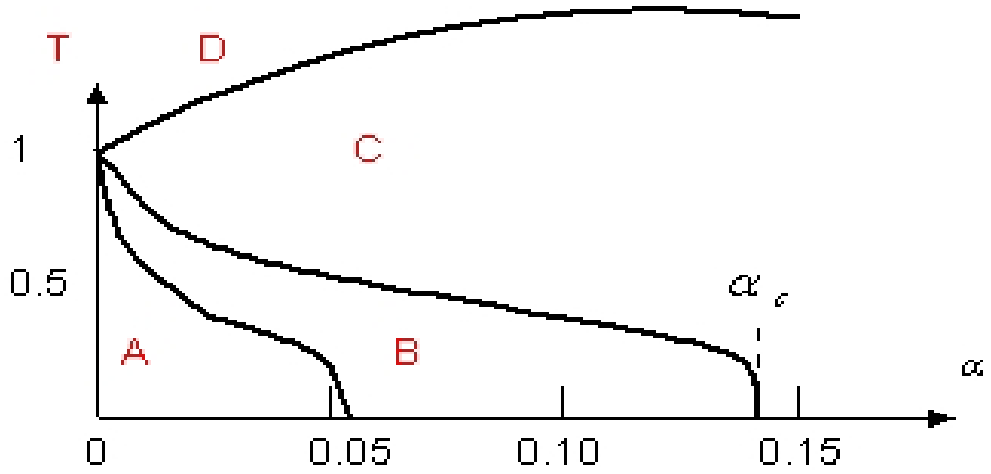
Where we have written m instead of m_1.

- We can find the capacity of the network by solving these three equations. Setting $y = m/\sqrt{2\alpha r}$, we obtain the equation:

$$y\left(\sqrt{2a}+\frac{2}{\sqrt{\pi}}e^{-y^2}\right)=erf(y)$$

- This equation can be solved graphically as usual.

Finally we can construct the **phase diagram** of the Hopfield model, which is shown in the next figure:



- We can observe the following:
  - There is a critical value of α where the non-trivial solutions ($m \neq 0$) disappear. The value is $\alpha_c \approx 0.138$;
  - Regions A and B both have the retrieval states, but also have spin glass states. The spin glass states are the most stable states in region B, where as in region A

the desired states are the global minima;

- In region C the network has many stable states, the spin glass states, but these are not correlated with any of the desired states;
- In region D there is only the trivial solution $<S_i>=0$;
- For small enough α and T there are also mixture states which are correlated with an odd number of the patterns. These have higher energy than the desired states. Each type of mixture state is stable in a triangular region like AB, but with smaller intercepts in both axes. The most stable mixture states, extend to 0.46 on the T axis and 0.03 on the α axis.

Conclusions

Conclusions

- The Hopfield network is a model of associative learning and it is inspired by the statistical mechanics of magnetic materials.
- There are many other variations of the basic Hopfield model. However, for all these variations the qualitive results hold even though the values of the critical parameters change in a systematic way.
- We can use the mean filed approximation in order to calculate the storage capacity of the network.
- The Hopfiled model can handle also correlated patterns using the method of *pseudo-inverse matrix*.

Conclusions-1

- The network can be used as a model of *Central Pattern Generators*.
- The model can also be used to store sequences of states. In this case the point attractors become *limit cycles*.