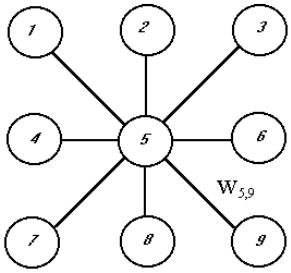
Associative Memory

- Mnemonists or “Memory men”
  - Can perform amazing feats of memory
  - Use the *Method of Loci* developed in Ancient Greece
    - Draws upon our exceptional ability to build spatio-cognitive maps
  - To memorise a list of items -
    - Think of a walk or route you know very well
    - Visualise successive items at places along the route
  - To recall the list of items
    - Take a mental stroll along the route retrieving the items as you go
- This is memory by association or Associative Memory

Hopfield Nets (1982)

- Due to John Hopfield
- Did much to restore the credibility of ANNs following Minsky & Papert’s book
- Hopfield’s key contribution was to provide an analysis of the network he devised in terms of the energy of the system
- His analysis was applied to many other types of ANN, such as Multi-Layered Perceptrons
- Hopfield Nets are associative memory devices

The Hopfield Network



- Each node is connected to every other node in the network
- Symmetric weights on connections ($w_{5,9} = w_{9,5}$)
- Node activations are either *-1* or *+1*
- Execution involves iteratively re-calculating the activation of each node until a stable state (assignment of activations to nodes) is achieved

The Hopfield Equations

- Training performed in one pass:

$$w_{ij} = \frac{1}{N} \sum_{k=1}^{n} p_i^k \, p_j^k$$

where,

w_{ij} is the weight between nodes *i* & *j*

N is the number of nodes in the network

n is the number of patterns to be learnt

p_i^k is the value required for the *i*-th node in pattern *k*

- Execution performed iteratively:

$$s_i = sign\left(\sum_{j=1}^{N} w_{ij} s_j \right)$$

where,

s_i is the activation of the *i*-th node

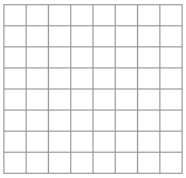
Analysis of the Hopfield Network

- Does the equation used to set the weights make stable states of patterns to be memorised?
  - The Generalised Hebb Rule ensures that this is the case
- Does the equation used to determine the activations of the nodes converge on stable states?
  - Hopfield's energy analysis enables us to prove that convergence occurs
- There has to be an upper limit on the number of patterns which can be memorised. What is it?
  - What is the memory capacity of an N node Hopfield network?

Hopfield Applications

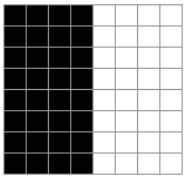
- Pattern Completion
  - Content Addressable Memory
  - Partial patterns can be completed to reproduce previously learnt patterns in their entirety
    - Incorrect patterns (patterns with errors or noise in them) can be corrected if they contain enough correct "bits" in partial patterns for pattern completion
- Combinatorial Optimisation
  - Learnt patterns are simply attractors
  - These are minima of some energy function defined in terms of the *wij* and *si* variables
    - Hopfield networks can be used to find the minima of the objective function of an optimisation problem if we equate the objective function to Hopfield's energy function and define weights, thresholds and activations accordingly

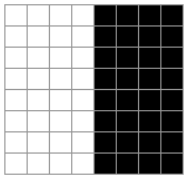
Pattern Completion (I)



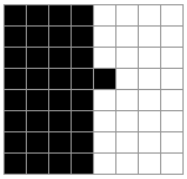
Learn this pattern

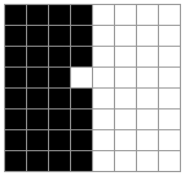
Inverse learnt too





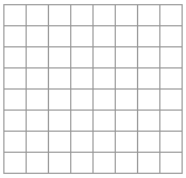
Present these two patterns and iterate ...





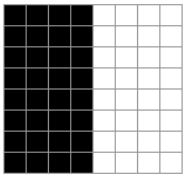
Both converge on top-left memory

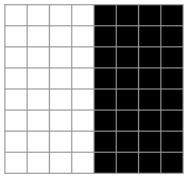
Pattern Completion (II)



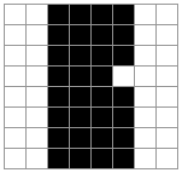
Learn this pattern

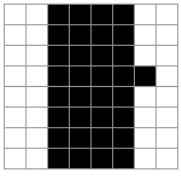
Inverse learnt too





Present these two patterns and iterate ...

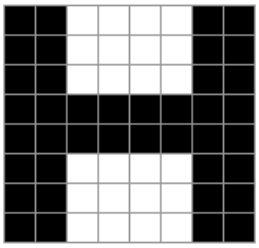




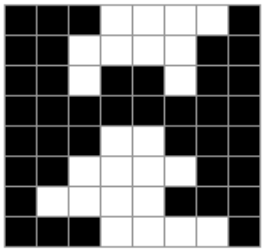
Converges on top-left

Converges on top-right

Pattern Completion (III)



64 pixel image of an “H”



Same image with 10 pixels altered
(I.e. approximately 16% noise added)

Pattern Completion (IV)

Food for thought - flight of fancy?

Memories of a deceased dog named Tanya ...

- Suppose we could create an associative network which enabled a small subset of the nodes below to trigger all of the other nodes
- Nodes could also make connections with other ANNS or (brain areas) so that the “Dog” node triggered the recall of “data” about dogs, etc. and vice versa

Dog	Lead	Slobber	Black
Mongrel	Dew Claws	Tanya	Brown Eyes
White Fleck	Dead	Wagging Tail	Wet-dog Smell
Hills	Cats	Floppy Ears	Snow

Combinatorial Optimisation

- The aim of optimisation is to find values for the input parameters of a system which yield a best, or optimal, output from that system
  - This normally boils down to minimising or maximising some cost, or objective, function of the input parameters
- Problems in which the input values are finite in number, or discrete, rather than infinite, or continuous, are known as combinatorial optimisation problems
- Hopfield Networks and, in particular, Hopfield's energy analysis offer a potential tool for solving non-linear combinatorial optimisation problems
  - The form of Hopfield's energy function limits the optimisations which can be addressed to those with quadratic objective functions - but this is not a huge limitation

Hopfield's Energy Function and Objective Functions

- Hopfield's energy function is a quadratic in the activations of connected nodes

$$H = -\frac{1}{2}\sum_{i\&j} w_{ij} s_i s_j$$

- If we can form a quadratic objective function, *O*, for an optimisation problem we can use it in place of Hopfield's energy function
  - Coefficients of the quadratic terms in *O* can become weights in the Hopfield network
  - Coefficients of the linear terms in *O* can become the thresholds of nodes in the Hopfield network

Optimisation Example (I)

- Task Assignment Problem
  - *x* people are to perform *x* tasks
  - The rate at which each person can perform each task is known (not all the same of course)
  - Determine the most efficient allocation of people to tasks
  - I.e. maximise the overall rate for performing all of the *x* tasks

	1	**2**	**3**	**4**	**5**	**6**
Anne	10	5	4	6	5	1
Brian	6	4	9	7	3	2
Clive	1	8	3	6	4	6
Dora	5	3	7	2	1	4
Edith	3	2	5	6	8	7
Fred	7	6	4	1	3	2

Optimisation Example (II)

- For the Task Assignment Problem a valid solution will have exactly one node in each row and column firing
  - First we relax the condition that node activations must be in {-1, +1}
  - We shall let them vary within [0, 1] by using a new problem-specific activation equation
  - A necessary (but not sufficient) condition for a valid solution is that the sum of all the activations should be *x*
    - We need to minimise $(overall\_sum\text{-}x)^2$
- Our objective function should seek to maximise -
  - For each person (row), the sum of each node activation multiplied by the associated rate
    - We need to minimise *row_sum*-1*
  - For each task (column), the sum of each node activation multiplied by the associated rate
    - We need to minimise *column_sum*-1*

Optimisation Example (III)

We now construct our energy function as a weighted sum of the three terms identified earlier -

$$A*(overall\_sum-x)^2 - B*row\_sum - C*column\_sum$$

- If we can determine suitable values for *A*, *B* & *C* then we can define a Hopfield Network which will minimise this expression and thus deliver a solution to our optimisation problem
- Note that *row_sum* and *column_sum* can be viewed as inhibitory inputs to a node *i* from the other nodes in the same row and column
  - We can therefore equate the rate at which person *i* does task *j* to our weights w_{ij}
- A change in the activation of a node changes the energy as defined by the above expression
  - The new activation for a node can thus be obtained simply by adding the result of the above expression (for an energy change) to the current activation
- Initial activations are set randomly

Optimisation Example (IV)

- A student project investigating the Task Assignment Problem gave the solution reproduced below after 73 iterations through the activation equation for each node
- The total rate for this solution is 44 which is, indeed, the maximum achievable

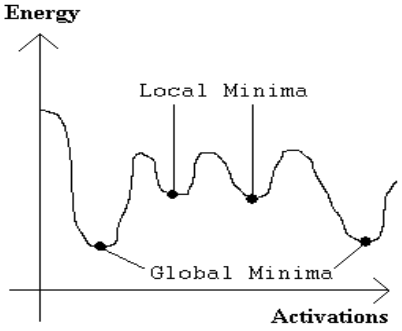
	1	**2**	**3**	**4**	**5**	**6**
Anne	**1.00**	0.02	0.02	0.03	0.02	0.04
Brian	0.02	0.06	0.07	**1.00**	0.04	0.10
Clive	0.03	0.19	0.07	0.10	0.06	**1.00**
Dora	0.03	0.06	**1.00**	0.06	0.05	0.13
Edith	0.02	0.04	0.04	0.04	**1.00**	0.10
Fred	0.04	**1.00**	0.08	0.08	0.07	0.15

More Optimisation Examples

Other classic examples include -

- Travelling Salesman Problem
  - Find the most efficient tour of *N* locations such that each location is visited just once
- Graph Partitioning Problem
  - Divide a graph into *K* partitions such that the number of connections between partitions is minimised
- Knapsack Problem
  - Given a utility measure and a space requirement for each of *M* items which might be packed into a knapsack which is not large enough to hold them all, select a set of items which will maximise the overall utility of the knapsack's contents

Local Minima



- We know each memory we store in a Hopfield Network also makes its inverse a memory - these are global minima of the energy surface
- Linear combinations of memories also become minima of the energy surface - local minima

Linear Combinations

- These “false” memories are of great concern because there are a lot of them -
  - Suppose we wish to store the following three memories

Pattern 1	*-1, +1, -1*
Pattern 2	*+1, +1, -1*
Pattern 3	*+1, -1, +1*

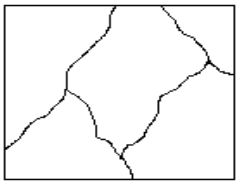
  - The linear combinations of these memories include

+1, +1, -1	*P1+P2+P3*
-1, -1, +1	*P1-P2+P3*
+1, +1, -1	*-P1+P2-P3*

    etc.
  - I.e. up to 8 of them here

Simulated Annealing

- In metallurgy the process of annealing is used to remove stresses from forged parts and reduce brittleness by heating followed by gradual cooling



Stress Lines

- Simulated annealing mimics this process by using stochastic nodes to “raise the temperature” (increase the energy) initially and then gradually reduce it

Stochastic Nodes

Instead of setting the activation of each node according to -

$$s_i = sign\left(\sum_{j=1}^{N} w_{ij} s_j\right)$$

we make the assignment to s_i dependent on a probability distribution -

$$P(s_i = \pm 1) = f_\beta(\pm net_i)$$

$$= \frac{1}{1 + e^{\mp 2\beta net_i}}$$

where β increases as the pseudo temperature decreases - $\beta \propto \frac{1}{T}$

Boltzmann Machines

- If the probability distribution used to assign activations to nodes is the Boltzmann-Gibbs distribution we have a family of ANNs called Boltzmann Machines
- Boltzmann machines are more than mere stochastic Hopfield Networks
  - They also incorporate hidden nodes (nodes which are neither input nor output nodes)
- Nor can their weights be determined in a single pass
  - They must be learnt as in a MLP
- Statistical mechanics is needed to fully analyse these ANNs