NEURAL NETWORK THEORIES AND THEIR SIMULATION

An investigation into Parallel (Neural) Network Theories, the problems of simulating and implementing the theories, and how the results of the implementations may help the understanding of intelligence and brain function.

MSc. Dissertation by P.Beautement.

Brunel University - 1984

10/4/92 Reprint

Re-edited July 2023 for PDF version

INDEX TO SECTIONS.

SECTION 1 - OUTLINE OF THE PROJECT

1. The suggestions by F.H.Crick and G.Mitchison on "The functions of dream sleep" have created a lot of interest, not least because it renews interest in autoassociative recall (networks which recall the whole of a stored pattern from a fragment), and in asynchronous, parallel (and self-organising) networks in general.

2. There are comparisons to be made between the validity and effectiveness of many types of network theories. In this project the associative network of the type described by Kohonen [1] and Hopfield [2,3], the subtractive network of the type described by Dobson [4,5], and the randomly connected network implemented in the hardware WISARD system [6,7] are implemented and studied.

3. All three networks were implemented on an IBM P.C. using IBM BASIC. This is because the Microneye camera used for some of the input of training and testing images works under this language. It would have been preferable to use U.C.S.D Pascal.

4. Once the networks were implemented, it was then necessary to assemble training sets which could be used to compare the performances of the networks. For the WISARD network the training sets were faces, bars and shapes. For the other two networks the training sets were two or three characters of text.

5. It was soon clear from testing the performances of the networks, that all the networks had various strengths and weaknesses. It is also clear that this work is really only a preliminary, and that there is a whole range of follow on work concerned with combining the networks in various ways. This is discussed in Section 9.

6. In the course of the project, and during background study and research, it has been necessary to consider the following points:

- The problems of network simulations in relation to models of brain neurophysiology. The significance of transformations that may occur in the brain at synapses and within neurons and axons (Section 3).
- The problems of serial simulations of hardware networks, and of hardware parallel network construction (Section 10).
- The implications of network models to "psychological" understanding of the brain (Section 11).

7. In conclusion, the project has provided a set of simulations on two discs with an introduction to the networks, which could be used by anyone wishing to familiarise themselves with the workings of three of the main types of networks being studied at the moment. It has also made clear the vast scope of research still to be done in this area and the potential for exiting and useful discoveries.

---------- 0 ----------

SECTION 2 - INTRODUCTION TO NETWORK THEORIES: A DIFFERENT APPROACH TO ARTIFICIAL INTELLIGENCE (AI)

8. During recent times three areas of human endeavour have come together more closely than might have been thought. Theses areas are neurophysiology and studies of brain function, psychology and artificial intelligence (A.I.), and studies of computer systems. As it becomes clear that computing machines are ever more useful tools, and as research advances the speed and utility while reducing the size and cost of the serial computer, computer specialists are lured by visions of "Intelligent Machines". Clive Sinclair is an example, who on Radio 4 recently spoke of robots and even human-like androids as being a likely development "soon". People like Sinclair look to A.I. for guidance, and to neurophysiology and psychology for models. But what has been achieved in A.I. towards reaching the goal of "Intelligent Machines" and reproducing human-like behaviour?

9. A.I. people (Boden [21]) would have us believe that a big enough computer program, running in a complex enough machine, would, over a certain threshold suddenly become intelligent. [Note that when I use 'intelligent' without quotes I mean intelligence as we understand it in humans.] The development of A.I., using computer programs to simulate human-like behaviour has intrigued psychologists, who have tried to explain the cognitive processes in humans in terms of programs running in the brain. A.I. in turn has looked (in a somewhat circular fashion it seems) to psychology for models to implement.

10. What of the other disciplines that I mentioned at the beginning of this section? Neurophysiologists, in tracing neural pathways, are beginning to map out the complexity that exists at even the "lowest" levels in the brain (human or other). They are also showing that the "processing" that occurs is not a serial, controlled process, but an asynchronous parallel process, where the behaviour we see comes from a complex juxtaposition of many factors, some known, and some only dimly hinted at. (Hubert & Wiesel [22], Kandel [22,27], Marr [28]). Computer scientists, though provided with a swift serial process, have a dilemma. At some point, in the pursuit of further speed, serial processes will have to be joined up in a parallel way and if this is to be done, how is it possible to control the exchange of information between the different sections? Systems with clocked next states are possible, but they are slower than the ideal. However the brain manages without a clock, so can the brain's processes be copied in some way?

11. For many years, progressing alongside the above work, there has been theorising about networks [19,20,31]. What does happen when you try and connect up something to be as complex as the brain? How could it be achieved, using what sort of "neurons" to make what sort of machine?

12. Psychologists following the information processing approach find that describing behaviour in terms of interconnecting boxes, each bit of which seems to be programmable, are faced with the daunting task of specifying so many interconnected boxes that nothing really seems to be explained, especially in the absence of any theory of how the boxes would work "en-masse". In fact some end up dismissing facets of behaviour as not of relevance because they are part of a different "system" (Warrington seminar '84). Neurophysiologists themselves, in describing and specifying larger and larger parts of the brain, are looking for patterns of connectivity and functional units which would obviate the description at the purely neural level. Attempts in this area (Marr [28], Hubert & Wiesel [22]) have made progress, but have shown that there is much scope for further work, and more importantly, a need for a good theory of the sort of units and their connectivity that might be expected.

13. The failures in A.I. to produce intelligence (but not failures in terms of useful tools) are beginning to produce what might be called a paradigm shift, that is, established workers (Winograd [10], also from outside, Searle [12], Palm [36]) are reappraising the aims and aspirations of A.I. This means that there will no doubt be similar moves in psychology.

14. What then do the development of better network theories offer to these disciplines? Hardware implementations such as WISARD can perform excellent pattern recognition without any program, and illustrate many of the possible advances that might come from better network theories. For example WISARD is very quick, versatile and robust, also it has shown potential not originally envisaged. However the hardwiring is (at present) very time consuming. Specifying beforehand exactly what a network may do, and explaining how it works is not easy. It may be that funding for research in the past has not always been forthcoming because of the apparent success of serial computers against what appears to be no more that extravagant hopes.

15. There are many sorts of networks which hint at different properties. I follow Luria's [29] concept of the brain as functional units. These appear to be different types of networks [27] connected together to extract and use the information obtained from the sensory input. There are problems in simulating or producing networks for experiment, and it is difficult to decide on criteria for comparison or efficiency. It is easy to make assumptions about the significance of different states within the network, which may well be wrong.

16. Automata theory gives a rigorous environment for studying some aspects of networks. However, the problems will be discussed in the next section.

---------- 0 ----------

SECTION 3 - NETWORK MODELS AND THE PROBLEMS OF NETWORK SIMULATIONS

17. As it is clear that there is value in studying networks, what is the best way of doing this? To construct a hardware implementation takes much time. Firstly because of the purely manual problems of making many thousands of small, fiddly, but of necessity, accurate, connections. Secondly because of the problems of testing or alteration. It would seem to be a better idea to make a software model that can be easily altered and used for experiment, so that promising arrangements can be followed up, and the others abandoned without the waste of a great deal of time and expense.

18. There are however some important problems to be overcome, and decisions and assumptions to be made as follows:

- What size of network represents a valid minimum for experimentation?
- What aspects of the network are going to be varied, and what weight of significance do these variables have such as:
- Connectivity, fully or partly, randomly or regularly?
- Resistance in connections, i.e. loss of information.
- Capacitance and field effects between different lines.
- What transformation or storage is to take place at the "neuron"?
- What transformation or storage is to take place at the "synapse"?
- What is the mechanism for the transformation?
- How is the storage to take place?
- To what extent does the long time taken by the serial simulation of a big network prejudice the experiments?
- To what extent is the simulation only an approximation of parallelism and the true properties of the network?
- How should the different states of the network be interpreted and related to any consideration of cognitive states?
- How could several networks be interconnected using further networks, rather than A.I. programs.

19. The solution to these problems is discussed in the paragraphs below. It is obviously necessary to make many assumptions, and so be able to vary as few variables as possible at any one time. The making of reasonable assumptions is obviously vital to the success of any experimentation.

20. Network Size. The size of network chosen for a simulation is usually taken as the smallest possible to show the properties of a particular network, and at the same time to run at reasonable speed. Most networks use arrays to record the strength or state of a particular "synapse" or connection. Arrays of more than about one to two hundred positions per side begin to be quite slow (> 5 minutes) in operation. The implementations in this project use 128 position arrays as the largest input/output vector. Also, is it possible to be sure that a small network represents truly the properties of a bigger network? Can one just extrapolate the results from a smaller to a bigger network? Comparisons between the 18 vector nets and the 128 vector nets in this project, would seem to indicate that such an extrapolation is valid. However, A.I. viewpoints are that "something happens" with larger programs, where there should be an emergence of intelligence. Are there properties emergent with large networks that would be missed with a small one? Undoubtedly there will be some.

21. Significance of Variables. What assumptions and decisions can be made to cut down the variables mentioned in para 18b(1)_to (7)? Taking each of the variables as follows:

22. Connectivity. Effects from variations in connectivity are known to be significant. The Dobson network works entirely on the basis of cutting links. But the effects of setting up networks with regular or random connections are more subtle. The brain is neither connected totally randomly nor with total regularity, so networks which function only when connected in a regular form are less robust and so less versatile. The Dobson and Hopfield networks in this project both use fully interconnected nets, with regular connections, though it is clear that the Dobson net will function with random connections and the Hopfield

net will not. The WISARD system uses random connections, and is not interconnected in the same way as the other networks. Unfortunately it has not been possible to experiment very much with the effect of variations in arrangements.

23. Resistance and Capacitance. Effects from resistance and capacitance are usually discounted in network simulation, although they cannot be discounted from silicon hardware implementations. The various field effects which can occur are discussed in [27] and need to be addressed.

24. Data Transformation and Storage in the Neuron. The "neuron" is taken as being the most significant transducer in the networks (as in the brain), and so usually attracts the most attention. In the subtractive net (Dobson), the simultaneous firing of two neurons as a result of the association between an input "image" and an output "image", causes the link between the two neurons to break (see section 6). In the additive associative net (Hopfield), the transformation is a more complex summation process and is explained in section 5. The WISARD system is difficult to see in neuronal terms, and is explained in section 7. However, in all cases, writers tend to compare the neuron to a single computer memory "bit". This is a serious mismatch. Each human neuron, with its many "processing" modes and high interconnectivity, equates with at least a powerful serial computer.

25. Data Transformation and Storage at the Synapse. Synaptic effects are usually taken to be purely excitatory or inhibitory. Recent neurophysiology [18,22,27,35] has shown quite clearly that there are many effects underlying whether a synapse releases neurotransmitter chemicals or not, and how much is released. From the above references and Luria [29] it is possible to suggest that many of the processes in the brain may be modified by purely synaptic changes. In fact the whole question of the role of brain and body chemistry tends to be simplified out of all existence by trying to explain cognition and brain function in pseudo-computer science terms. Some of the chemical and synaptic effects to which I am referring are listed below as follows:

- Changes in cortical tone.
- Alteration of the frequency of nerve cell firing.
- Amplification effects by cyclic AMP in the post synaptic terminal.
- Summation of impulses on post synaptic terminal by the accumulation of chemicals.
- Inhibition of transmission of a synapse.
- Release of neuropeptides causing a whole range of physiological changes, e.g. thirst etc. and possibly affecting such functions as learning.
- Type of synapses, quick or slow acting on the post synaptic cell.
- The role of rapid and slow transmission through different types of nerve fibre.
- Some transmitters are excitatory in one part of the brain and inhibitory in another.
- The fact that there are synapses not just on the "next cell" but also on its own cell body, on axon terminals, on its own and other dendrites and on itself.
- Certain sections of the brain are not shielded and specifically respond to blood borne hormones.
- The role of the glial cells may yet be of interest in relation to nerve messages.

The comparison of the neurological validity of the different network theories are discussed in section 8.

26. Transformation and Storage Mechanisms. The mechanisms for the transformations in the neurons and for the storage of the "information" are easy to explain in mechanistic terms in the simulations, but as has been illustrated above, more difficult to explain in relation to the brain (see section 8).

27. Serial Simulations. The slowness of the serial simulations discourages experimentation on larger networks, and so opportunities for discoveries may be missed, especially in the area of interconnecting different types of networks.

28. Simulation Validity. There is the question of whether it really is possible to be sure that the serial simulation accurately represents the functioning of a parallel hardware network. It may not matter much at this stage, but it is certain that subtleties of networks will be missed. Section 9 discusses the possibilities for combining networks to exploit their relative properties.

29. Interpretation of Network States. The interpretation of network states is an important question which has been highlighted by the work of Crick [9]. How should the stable states in networks be interpreted? The states (in automata theory terms) whether confluences or transients, or whether part of an ergodic cycle (see Kauffman [17]) can be interpreted in different ways. In section 11, the problem of interpretation of network states is discussed in more detail. In interpreting the output of a network, as a reflection of its states, it is necessary to have some form of presentation which allows the experimenter to easily make judgements about the effectiveness of a particular network.

This is clearly shown in Crick's demonstration program (HOPDEMPB) where the input and output vectors are strings of 0's or -1's and 1's. Even though he provides information about the "distance" of one vector from another, it is not easy to see any patterns of change, success or failure. In the 18 and 128 vector Hopfield and Dobson programs implemented in this project, the input and output vectors are represented as letter characters, each part of the vector being one of the pixels that makes up the letter. In this way it is easy to see what degradation of the input (i.e. loss of integrity of the information) has taken place, patterns are easier to locate and so general properties of a particular network can be quickly established. This subjective assessment is then followed up by quantitative measurements. The WISARD system uses the Microneye digital camera for input, for the same reasons.

30. Network Interconnections. Network interconnection is discussed in section 9, where the possibilities for using networks to relate and connect other networks and so avoid A.I. type controlling programs are considered.

---------- 0 ----------

SECTION 4 - INTRODUCTION TO THE NETWORKS AND ASSOCIATED PROGRAMS

31. The three network programs and their associated data files are on the two discs A and B (for the IBM PC) provided with the dissertation. Disc A is a system disc, and when booted will take the user straight into a program called NETCHAIN. This program consists of an introduction, instructions and then a menu of the programs that can be selected. On the following two pages there is a sample of the information that appears on the screen during this program. Also on disc A, but not appearing in the menu, are three supplementary programs as follows:

- LAYOUT is used to check that the random numbers used in setting up the input connections for the WISARD network are distributed between the two "discriminators" used in this system (see section 7).
- MAKERAND is used to produce a set of pseudo-random numbers which the above program uses (see section 7).
- RAMSTATES allows the state or memory of the WISARD system to be examined after a training session to see if the memory is saturated (see section 7).

These programs can be run by exiting to IBM BASIC, loading them (use function key F1 and the program name), and then press the RUN key (function key F2).

32. Listings of all the programs are in the Appendices at the end of the dissertation, with the training sets used to produce the example outputs in sections 5,6 and 7.

33. For the 18 and 128 vector Hopfield and Dobson networks, training sets and test patterns are input from the keyboard as characters. For the WISARD network a Microneye camera must be connected to the computer. It was originally hoped to use the camera for all three, but the arrays for the subtractive and associative networks would have been 512 x 512, producing a long running time (about 40 minutes between states) and would have used 262K of memory. (Not available at the time). As the comparison between the subtractive and associative nets are the most direct, it was important that they should have an identical input form, and be of identical size. The letter input did not provide enough variety of input for the WISARD network, and so the camera was used.

34. The following three sections of the dissertation each have an identical layout as follows:

- A description of the type of network being studied.
- A description of the programs used to implement the net.
- Explanation of the instructions for using the program.
- Discussion of the example tests and training sets.
- Synopsis of the network's general properties.

---------- 0 ----------

SECTION 5 - THE ADDITIVE AUTOASSOCIATIVE NETWORK (HOPFIELD TYPE)

THE NETWORK - DESCRIPTION

35. This network owes its origins to, and has some similarities to, the Perceptron and follows in the line of the McCulloch and Pitts model using a Hebb like synapse, although it differs significantly from the linear associative nets (see [1,2,3,30,37].

36. The basic concept is of a neuron, part of a group of N neurons in the network, with N-1 connections through synapses from any one neuron to the rest. Each neuron, i, has two states, firing ($V^i = 1$) or not firing ($V^i = 0$). When neuron i has a connection made from neuron j, then the strength of the connection is defined as T^{ij} (neurons that are not connected have $T^{ij} = 0$ as neurons are not connected to themselves). The instantaneous state of the system is specified by listing the N values of V^i, i.e. a list of firing or not firing. The state changes in time as follows. Each neuron has a threshold U^i, and the neuron readjusts its state randomly (but at a mean rate W) according to whether it is above or below the threshold ($U^i = 0$).

37. In this model there is no directionality in the network, the connections and flow of signals can go back and forth through the network. In theory the network is asynchronous, and so over time the network reaches stable limit points X^a,X^b,..... For any state vector X sufficiently near X^a,X^b,.. the system flows towards these stable states, and in this sense has a formal similarity to many other models of physical systems, e.g. stable magnetic orientations and domains, N Ising spins, genetic models, (see [17,19,20,31,38]).

38. To store information, changes occur in the weighting of the thresholds, T^{ij}, representing the sum (from the set of states V^s, s = 1..n) of multiplying term ^{i} by term ^{j}. This point can be more clearly demonstrated by running the program HOPNET18, and looking at the synapse table printed out as each new vector is put in; i.e. the program takes term ^{i} of the input vector, multiplies it by term ^{j}, and then adds it to the value in the matrix at position T^{ij}.

39. To retrieve information, term ^{i} of the test vector is multiplied by the value in the matrix at T^{ij}, and the value is added to the value in the output vector at position ^{j}. The output vector then becomes a set of positive and negative values, reduced by the threshold U^i, to a binary word of N bits. In the demonstration provided by Crick, this output vector is recycled as an input vector till the network settles to a stable state X^a. This is not done in the 18 vector simulation, but is available in the 128 vector simulation.

The neurological validity of the model and its effectiveness are discussed in section 8.

DESCRIPTION OF THE PROGRAMS HOPNET18, HOPNETPB AND HOPDEMPB

All the above three programs use the same algorithm for the calculation of thresholds and output vectors. For N neurons, and an input vector of length N, the program specifies an input array and output array of N terms, and a threshold matrix of N x N terms. Initially the values in the threshold matrix are all set at 0, but they could be set to a set of values representing a set of states already stored. This can be done with the program HOPNETPB, and so saves about 40 minutes of laborious teaching time. To calculate the threshold values from the input array (VECTOR) to the threshold array (SUM), the program follows the form:

```
FOR I = 1 TO N
FOR J = 1 TO N
temporary value = VECTOR(I) * VECTOR(J)
SUM(I,J) = SUM(I,J) + temporary value
NEXT J
NEXT I
```

To calculate an output vector (OUTPUT) from the test vector in (INPUT) and the synapse weights in (SUM), the program follows the form :

First empty the output array then :

```
FOR I = 1 TO N
FOR J = 1 TO N
OUTPUT(I) = OUTPUT(I) + (VECTOR(J) * SUM(I,J))
NEXT J
NEXT I
```

To recycle the output as a new input vector it is then only necessary to read the values from OUTPUT into INPUT, and reset the output to 0. Once the output has stabilised, OUTPUT is read off.

IF OUTPUT(I) > 0 THEN output a one
ELSE output a nought.

40. The HOPNET18 and HOPNETPB programs get their input vector from the POINT(x,y) BASIC function. On the graphics screen an illuminated pixel is seen as a 1 and a black pixel is seen as a nought. The 0's are changed into -1 for the multiplication. The HOPDEMPB program gets a built in pseudo-random input, but I have modified this program by Crick to allow the user to input their own test vectors at the end of the program. There is no "unlearning" in any of the programs (see section 11).

41. The HOPNET18 program produces only half of the synapse table as this is symmetrical. The HOPNETPB and HOPDEMPB programs produce a full size table, and so are much slower programs because of this and their size.

EXPLANATION OF INSTRUCTIONS FOR USING THE PROGRAMS

42. Instructions for loading the programs from the discs A and B are to be found in Section 4. On the next eight pages will be found sample outputs, showing what will appear on the printer when the programs are run as follows:

43. First Page - This is the output from HOPNET18. The 18 vector programs only read the pixels from the third line of the characters (this allows lower case letters to be used), and so in the input vector a -1 represents a blank pixel, and a 1 represents a white pixel. The example shows the synapse table after learning three vectors. In the testing memory stage, the test vector and the output vector are shown. A longer test session where 30 patterns were learnt can be found in Appendix D.

44. Second to Fourth Pages - This is the output from HOPNETPB. Note that the input is now the whole of two letters. They are enlarged on the print-out so that the position of the pixels can be seen. It takes about a minute for each vector to be learnt. In the testing stage the test vector and output vector are both printed for comparison. Note how easy it is to see the autoassociative (or holographic) effect of the memory in recall of LL from LE, compared to the same effect occurring in the HOPNET18 program in recall of the EEE from EE. Typing in RDM will cause a random test vector to be produced. Further cycles of testing can also be selected. The options menu in this program also allows the following:

- The existing state of the synapse table can be stored on the data disc, allowing an experimental set to be learnt and then saved for further testing. This has already been done, ten patterns (set A) have been stored under the file name MEMHPL10 and twenty (set B) under the file name MEMHPLET.
- The memory on disc can be loaded into the synapse table, for use by the network. Note that this operation will erase any existing memory state in the network.
- The synapse table can be printed out. A small sample has been included on the fourth page, showing values after learning LL and O0.

45. Fifth to Eighth Pages - This is the output from HOPDEMPB which appears on the screen and is fairly self-explanatory. If you wish to have the output to the printer, press the < Ctrl > and < PrtSc > keys at the same time, press them again to stop the printing.

46. This program gives the opportunity to experiment with some of the variables discussed in section 3. It is interesting to note that Crick, in several places, uses the phrase "it is best to have...". Some of the

limitations that he is hinting at are a result of the implementation, for example, that the neurons should change state asynchronously, or as he puts it "at random".

DISCUSSION OF EXAMPLE TEST AND TRAINING SETS.

47. The claims made by Hopfield in [2], viz:

"Additional emergent collective properties include some capacity for generalisation, familiarity recognition categorisation, error correction and time sequence retention."

48. The properties of the net are apparent, and can be seen by studying the test sets provided. The training sets: set A (10 patterns learnt and the synapse table stored in file MEMHPL10) and set B (20 patterns learnt and the synapse table stored in file MEMHPLET), are to be found in Appendix J, with part of the synapse table for each set. The following 17 pages show the results of two test sessions, one with set A and one with set B.

49. The random inputs show an interesting feature of this network, namely that the noise tends to be eliminated, and the output reflects generalised elements of the training set. The experiments with set B show that this generalisation may appear as an inverse image. The test input when characters are used rarely produces an exact output. There is usually some generalisation in set A (the EE is a notable exception), and a great deal of generalisation from set B (as there have been more patterns learnt). Attempts at stimulating holographic type recall, by giving FF for EE produce poor results. Instead what happens is a "rounding down" to the common generalisation, worse again in set B.

50. In the character input programs the ratio of on/off pixels averages out to around 50%, and as there are many parts of the character patterns that overlap, this might be thought to give an unrepresentative output. But the same trends are seen in the Crick demonstration, where the input is infinitely variable.

SUMMARY OF THE GENERAL PROPERTIES OF THE NET.

51. The general properties of the net are as follows:

- A large number of patterns may be learnt, only at the cost of generalisation.
- Hopfield states that when more than 0.15N states are learnt (this would be 19.2 patterns for the 128 vector input) serious errors in recall start to occur.
- Autoassociative memory recall does occur, but the output is rarely accurate unless only a little of the information is missing.
- Categorisation, i.e. grouping into like forms, is an aspect of the generalisations. With distinct enough classes categorisation is possible, but at the cost of less "information" in the input vector.
- Generalisation which occurs in this net certainly would be reduced by a good "unlearning" process.

---------- 0 ----------

SECTION 6 - THE SUBTRACTIVE NETWORK (DOBSON TYPE)

THE NETWORK - DESCRIPTION

52. Interest in this type of network stems from the fact that it is extremely plausible from the neurological point of view. The net operates by selectively cutting inhibitory links, which does not need sophisticated threshold logic, and which can function reliably in randomly connected circuitry, under various conditions (see Dobson [4,5]).

53. The main concepts in this network theory are as follows. An input group of N neurons are fully and randomly interconnected (N connections from each neuron) to an output group of neurons (not necessarily N of them). The neurons can either be in a firing or not firing state, as can the synapses. However, the connections from the input to the output array are directed, and so the synapse on the output array from the input array follow the state of the input cell (see the next page for explanatory diagrams).

54. To store information, there is an input vector in the input array and an associated output vector (which can be the same as the input vector) in the output array. If a term, ^{i}, in the input array has an excitatory value at the same time as a term, ^{j}, in the output array, then the link, L^{ij}, will be cut. That is, a firing neuron in the input array connected to a firing neuron in the output array will cause the link between them to be cut.

55. To retrieve information, i.e. when there is no potential input to be learnt, the output array reverts to spontaneous firing (or quiescence). A test vector at the input array, causing some of the neurons to fire, will have the effect of inhibiting the neurons in the output array where there is a connection. The output vector will therefore be a function of the input test vector and the patterns previously learnt. A test pattern bearing little resemblance to a pattern already learnt will have the effect of entirely inhibiting the output array. A test pattern very near, or identical to a pattern already learnt will have the effect of inhibiting only those cells in the output array to which it is connected, and so switching off all the cells but the ones that represent the correct associated output pattern.

56. As there are possible variations in the combinations of spontaneous excitation/inhibition, these have been discussed thoroughly in [4], along with the limitations of the network and its neurological validity (see section 8).

DESCRIPTION OF THE PROGRAMS DOBNET18 AND DOBNETPB

57.

58. These programs are similar in the presentation to the user to the HOPNET18 and HOPNETPB programs. The algorithm, of course is different. The program has an input array (VECTOR), and an output/association array (OUTPUT) both of N terms. There is a synapse table (SUM) of size N x N, which holds values only of 1 or 0, showing connection or disconnection. In the initial state the input and output arrays are empty and the synapse table is filled with 1's to show a virgin, fully connected net.

59. To store patterns of vectors an input vector is put in VECTOR and the associated output pattern is put in OUTPUT. The program then establishes the links to break as follows:

```
FOR I = 1 TO N
FOR J = 1 TO N
IF VECTOR(J) not exited THEN NEXT J
ELSE IF OUTPUT(J) not exited THEN NEXT J
ELSE [ there is associated input and output ]
SUM(I,J) = 0 [ so break the link ]
NEXT J
NEXT I
```

60. It is clear that a totally firing input array and output array would wipe out all the connections at once! However, Dobson has studied this problem carefully to come up with an optimum of about Log base 2 N of

the input array firing, that is about 7 out of the 128 terms in the input vector. The random input in the program puts about 64 firing cells in the input vector.

61. To test the network and obtain output, the input array is emptied and the output array is set to all 1's, i.e. spontaneously firing. To see which cells in the output array should be inhibited the program proceeds as follows :

```
FOR I = 1 TO N
FOR J = 1 TO N
IF VECTOR(J) not exited THEN NEXT J
ELSE IF SUM(I,J) < 1 (disconnected) THEN NEXT J
ELSE [ input activity will inhibit output cell ]
OUTPUT(J) = 0 [ so switch it off ]
NEXT J
NEXT I
```

The output array is then expressed directly, as firing or not, there is no need to further check thresholds. The synapse table for this network is not symmetrical and could be held in a program as a binary word, also because of the IF statements the program is significantly quicker in execution than any of the other implementations.

EXPLANATION OF THE INSTRUCTIONS FOR USING THE PROGRAMS

62. The input to these programs is again from character input. A difference to be noted is that during the teaching phase the input vector is not associated with the output vector automatically. The program therefore prompts the user for the associated output vector (which can be the same as the input vector).

63. The instructions for loading the programs from the discs are in section 4. On the following five pages will be found sample outputs, showing what will appear on the printer when the programs are run as follows:

64. The first page shows output from DOBNET18. Two patterns were learnt, and the resulting synapse table is printed out. A single test vector is then given and the output vector is shown as before. A longer test session in which 30 patterns were learnt can be found in Appendix G.

65. The second to fifth pages show the output from program DOBNETPB. The options allow the user to teach the network some patterns, test the network with a random test vector by typing in RDM, or with a character test vector.

66. The program options allow the following:

- The state of the network links can be saved on disc after a learning session. This has already been done by teaching 10 patterns,set C, and saving the state as file MEMDBL10. Also 20 patterns were taught separately, set D, and the state was saved as file MEMDBLET.
- The saved states can then be loaded into memory to continue an experimental session. Note that this will erase the existing memory.
- The synapse table for the state of the network existing at the time can be printed out.

67. Note that the use of the < Alt > key, with the number pad on the right of the keyboard allows the user to test the networks with a variety of characters. Unfortunately they do not appear as individual characters on the printer, so these characters are best used with programs HOPNETPB and DOBNETPB, where the user obtains the large print-out. For further details see the character set in the IBM manual.

68. There is no recycling of the output vector back in as an input vector with this network, as there is no flow through a state space. The program is also much quicker.

DESCRIPTION OF THE EXPERIMENTAL TEST AND TRAINING SETS

69. The claims for the accuracy, robustness and efficiency of this network made by Dobson are verified by experimentation with the network. Certainly the accuracy of recall is impressive, but as with the other networks there are distinct limitations and boundaries beyond which the network breaks down.

70. The training sets provided are: set C (10 patterns learnt and stored in the file MEMDBL10) and set D (20 patterns learnt and stored in the file MEMDBLET). The training patterns used in these sets can be found in section 15, Appendix J, with part of the synapse table for each set. The next 13 pages contain the outputs from two test sessions, one with set C and the other with set D.

71. The first thing to note is that the random test vector never produces an output, even from the net taught set D, which had obviously become saturated. The testing session on set C, which had obviously nearly saturated the net, produced consistent output. There is little generalisation or error. However, the test session on set D, showed the problem of a saturated net. Because the output vectors were all lower case there was little differentiation of the output pattern. In this case the net was performing as a pattern recogniser, i.e. the response is effectively saying "I've seen this pattern class before", but little else.

SUMMARY OF THE GENERAL PROPERTIES OF THE NETWORK

72. The general properties of the network are as follows:

- The network recalls the associated output vector with a high degree of accuracy till saturated.
- Saturation of the network still does not produce an output for a random or unknown input. The network then behaves as a class recogniser.
- The number of patterns that can be learnt is a function of the ratio of firing to not firing cells in the input vector. On the basis of the 64 to 128 that we have here about 2 patterns would start to saturate the net. On the basis of 6 to 128, something like about 70 patterns could be learnt.
- The network shows autoassociative recall, that is holographic like properties.
- Categorisation does not occur, except when saturated, but only into the categories of seen/not seen.
- Generalisation does not normally occur, the output is very much all or nothing.

---------- 0 ----------

SECTION 7 - THE SINGLE LAYER RANDOM NETWORK (WISARD TYPE)

DESCRIPTION OF THE NETWORK

73. This network was developed primarily as a basis for an integrated circuit hardware device. It has its origins in the development of neural models designed to exploit novel solid-state architecture (see [6,7]). In certain aspects it shows more similarities with the Dobson net than the Hopfield net. There is no need to store analogue or highly multivalued data.

74. The basic concept is of a memory unit with inputs through N "address" terminals, and the output a single binary terminal. Thus for an input vector X across a network of memory units M, the system performs a mapping

M X ----------> { 0,1 }

where M is a binary input vector of the form (m0,m1....m2 to the N-1). So M^j is a single bit word addressed by the ^{j}-th pattern from X.

75. The basic definitions are as follows:

- Within the system a discriminator is a group of K RAMs.
- A number of classes of patterns, C, can be specified (two in this implementation), this implies the existence of C discriminators.
- R is the number of bits in the image to be recognised (1024 in this example).
- N is the number of address (synaptic) inputs to each RAM, known as N tuple (this example is 4 tuple).
- A discriminator is usually made to cover the image just once, and so K (the number of RAMs) is R / N. The connections from the RAMs to the input image are made at random, but remembering that each pixel of the image has only one connection made to it.

76. To teach the network an image a pattern is presented at the input array of R bits, the memory locations within the RAMs are cleared to 0 before a new class is taught. In the simplest case each bit is then on or off, and so sends a 0 or 1 to each of the address terminals. For each RAM, there will then be a N bit word, eg 0100 or 1110 in this case, which locates one of the 2 to the N (16 in this example) locations within memory. A logical 1 is written to this location. At the end of a training set the RAM memory locations then contain 0's and 1's in relation to the number of patterns taught (see [6 p250]).

77. To test the network, a test pattern is presented at the input array. It is important to note that this Level 1 structure (see [7] for discussion of Level 2,3.. arrangements) does not produce output patterns like the other networks. In fact the Level 1 structure is concerned with the level of response of the C discriminators to a test pattern.

78. The test pattern, again R on/off bits, produces at each RAM an N-bit word, which specifies a memory location as before. In this test situation the bit stored in this location (0 or 1) is added to the bits located in all the K RAMs of a discriminator. The response from each discriminator is then an integer c, such that 0 <= c <= K. The response can then be quoted as a percentage of K. The responses from the different discriminators then define a response vector.

79. The confidence that a pattern belongs to a particular class can be specified in several ways (see [6 p247]). In this example a histogram is displayed proportional to the value of c for each discriminator. A percentage confidence of the difference between c1 and c2 is then displayed. The possible neurological interpretations of this network are discussed in section 8.

DESCRIPTION OF THE PROGRAM MINIWSRD

80. There are three components to this program, a fairly lengthy user interface section with an options menu, the section which deals with the control of the Microneye camera through assembler routines and the network implementation itself. The listing of the program is to be found in Appendix I. The description of the options menu is made in the next part of this section. The Microneye camera routines can be examined by referring to the excellent handbook.

81. As with the other network implementations there is a fairly lengthy initialisation sequence when the program is started. With the MINIWSRD program however, this is not just a matter of emptying the memory array. Each discriminator in this program is made up of 128 4 tuple RAMs. This means that each discriminator covers 128 x 4 = 512 bits of the 1024 input array. Limitations on memory and speed meant that a group of 256 4 tuple RAMs could not be used. The input array, which is a section of memory used by the Microneye camera, therefore needed to be divided up. Half allocated to discriminator one, and half to discriminator two.

82. The Microneye camera system places a topographically correct image in the 1024 positions in memory (64 rows of 16 columns), so by allocating addresses 1,3..15, 18,20..32, etc to discriminator one, and addresses 2,4...16, 17,19..31, etc to discriminator two, the input image is covered evenly. The output of program LAYOUT, (see Appendix B) shows this point clearly. The addresses so divided, now must be allocated to the input addresses of the RAMs of each discriminator at random. The program MAKERAND, (see Appendix B), does this, opening a file called Random. When MINIWSRD is initialising it uses this file to set up the "connections" to the computer's memory locations. Once these connections are set up they remain constant, and every time the program is loaded it uses the same file, i.e. MINIWSRD is "hardwired".

83. When the network is being taught patterns from a training set, each discriminator is taught separately, but the algorithm is the same:

```
FOR K = 1 TO 128 [ the number of RAMs ]
ADDRESS = 0
FOR N = 1 TO 4 [ the tuple ]
[ ADDRESS is the position in the RAM's memory ]
ADDRESS = ADDRESS + value found at the position in
input array where this tuple is connected
NEXT N [ this cycle gives a decimal digit which specifies an
array location ]
DISCRIMINATOR (C (RAM(K(ADDRESS)))) = 1
NEXT K
```

The testing of the network requires that a response be obtained from both discriminators. The program proceeds as follows:

```
[ set response values (RESPONSE1, RESPONSE2) to 0 ]
FOR K = 1 TO 128
[set location pointers (ADDRESS1, ADDRESS2) to 0 ]
FOR N = 1 TO 4
ADDRESS1 = ADDRESS1 + value at the position in the
input array where this tuple is connected
ADDRESS2 = ADDRESS2 + value at the position in the
input array where this tuple is connected
NEXT N
RESPONSE1 = RESPONSE1 + DESCRIM.ONE(RAM(K(ADDRESS1)))
RESPONSE2 = RESPONSE2 + DESCRIM.TWO(RAM(K(ADDRESS2)))
[ add up the values found in the ram memories ]
NEXT K
```

Next the confidence is calculated by taking the absolute difference between RESPONSE1 and RESPONSE2, and taking this value as a percentage of the larger response.

84. The results are displayed as pictures and histograms on the screen, and on the printer, (see later in this part for some examples of the output. The input array from the Microneye camera gives in effect in input vector to each discriminator of 512 bits, as opposed to 128 bits for the HOPNETPB and DOBNETPB. The actual storage is 128 x 16 = 2048 bits.

85. The content of the RAMs can be printed out using the RAMSTATES program. This shows the tuple wiring locations allocated to each RAM as well as the bit values in the RAM memory.

EXPLANATION OF THE INSTRUCTIONS FOR USING THE PROGRAM MINIWSRD

86. The next page contains a print-out of the screen display during the initialisation and early stages of running the program. The initialisation and loading of the random data file occur while flags appear to show the user that something is happening. As this program uses a great deal of memory, it is possible to get an out of memory error message after using the other programs. In this case it is best to reboot the whole system from scratch.

87. The Mini Wisard Activity and Setup options then appear. In any of the modes where there is the picture provided by the Microneye camera in the top left corner of the screen it is possible to use the Real Time Commands summarised below:

- Key 1. This puts the network into the test mode. Have the test set and camera ready before choosing this mode.
- Key 2. The use of the camera is simple. Note that it is best to press <2> (setup options) and then <b> (adjust camera), before anything else. The aperture and focus on the camera will then need to be adjusted. The camera is very sensitive to intensity changes, so a constant light source is necessary. Other setup options under key 2 are as follows:
- Feedback. Not available!
- Adjust Camera.
- Load Previous Memory. Note that when a new memory is loaded **the previous one is erased**. There are three memory files already on the data disc, MEMFACE, MEMBARS and MEMSHAPE, which can be used for further experiment.
- Save Present Memory. After a training session, the memories in the RAMs can be stored. This can save a great deal of time. The disc in drive B must have at least 18000 bytes free.
- Teach from Disc. As it is possible to save pictures, the system can then be left to learn the patterns unattended. 25 input patterns are learnt at a time, taking about 20 minutes.
- Clear Discriminators. This clears out any sets of patterns already learnt, leaving a blank memory.
- Save Class to Disc. The disc in the drive must have at least 27000 bytes free. This mode allows 25 patterns to be stored. It takes about 5 seconds per image. The camera and training set must be set up beforehand.
- Keys 3 and 4. These put the network in the learning mode. Patterns are learnt from the camera at this time (not from disc), so have the camera and training sets ready.
- Key 5. This puts up the real time command summary for reference. It is not possible to operate the camera or network in this mode.
- Key 6. This exits to the networks menu, or to BASIC if MINIWSRD has been loaded directly from disc.

DESCRIPTION OF THE EXAMPLE TEST AND TRAINING SETS.

88. The properties of this type of single layer net have been tested and described (see references in [6,7])The main areas of success and application are: (from [6])

> "...in image recognition tasks that are important in robotics ...as it can take decisions on 512 x 512 pixel images at the rate of 5 - 10 per second. Robotic vision tasks require decisions at this order of speed on

rotated or translated versions of objects, and sometimes demand the recognition of combinations of objects even if occlusion takes place."

89. The hardware implementation can certainly do these things, but does the program here? The main strength of this network if that it can be taught a particular object in any position or orientation in the input array, and can generalise to positions that are an approximation of positions seen.

90. The training sets used for the examples are found in Appendix K, and are three sets of two classes. Set one is Faces, two is Bars and three is Shapes. These have already been taught and saved as MEMFACES, MEMBARS and MEMSHAPE. Within each set are two classes. From the example in the Appendix, the distortion caused by the camera can be noted.

91. On the next 9 pages are examples of the results of a test session, which also involved a "random" input. The stimulus for the "random" input is to be found in Appendix K. It can be noted that the response of the Shape discriminators is low, because they were taught on an image containing most blank space, whereas the response of the others is higher, as their images have more dark areas. Note the difference in confidence between recognising the faces and shapes as opposed to the bars. The network is recognising two elements of the patterns as follows:

- The areas of the objects in absolute pixel terms, and
- The difference of distribution of the area between the objects.

Therefore objects that differ in area and distribution will produce a higher confidence of response than objects of the same area, but different distribution.

SUMMARY OF THE PROPERTIES OF THE NETWORK

92. The properties of the network are as follows:

- Pattern recognition occurs, but the network does not output an example of the pattern.
- An extremely large number of patterns can be learnt, but for only a limited number of classes.
- The network is not autoassociative.
- Categorisation occurs through training separate classes.
- A measure of generalisation can be obtained from the confidence figures, which can measure rotation etc.

---------- 0 ----------

SECTION 8 - COMPARISONS OF THE PERFORMANCES OF THE THREE NETWORKS

If the are compared and contrasted, it can be seen how they complement each other in a way which will give some insight as to how they could be combined to produce higher level properties.

NEUROPHYSIOLOGICAL COMPARISONS

93. Dobson Network. Evidence concerning changes in the brain during ontogeny [22,27] clearly shows that the brain cells, before birth, develop many thousands more synapses than are found in adulthood. In fact a large percentage of this excess is lost in the first five years of life, when most of a human's learning is done. This would seem to strongly support Dobson's theories, if these synapses are being destroyed as part of the learning process. Work by Alkon [18] has shown that in aversion the simultaneous association of a stimulus with an aversion can indeed switch off a synapse, but also he notes that the process can be reversed. However, physiological data has shown that an inactive synapse does in time decay, as the action potential with its associated neurotransmitter and other chemical activity helps to maintain a healthy synapse. All this gives Dobson's network strong validity. What though of the problems. If the Dobson approach is to be the only learning technique, are there enough synapses in the brain, remembering that some parts of the brain may never have synapses cut by such a process, and that at least half the connections must remain to avoid saturation? It is almost impossible to estimate an answer. What length vector represents any event we might want to remember? An alternative approach is to consider that this technique would not work for short-term and working memory, it is too slow. Also in some ways it is too exact. If we want to remember a telephone number for a long time, then this is the method; but to remember, associate, categorise and pigeon-hole similar events or objects this sort of memory is not satisfactory.

94. Hopfield Network. The Hopfield network would satisfy these criteria, but suffers from some neurophysiological implausibility. How would a synapse store -34, or 61 as a threshold weighting? Poggio in [39] considers some synapse arrangements for simple mathematical style operators. As has already been mentioned, firing frequencies, chemical amplification effects and "summing" by accumulation of neurotransmitter do occur. It may well be that for short terms (up to say a week) such a mechanism is in fact viable. Neurochemical and psychological work has been able to show (though not yet with any exact predictive accuracy for complex states) that certain brain "states" are associated with either chemical imbalances, or deliberately induced by hormones, eg adrenalin. The transition from the short term network learning to synapse inhibiting learning may well occur when we sit down to "learn this by heart".

95. WISARD Network. Before any comparison can be made, it is necessary to try to consider the WISARD network from the same neurological viewpoint that can be used for the consideration of the Hopfield and Dobson networks. The RAM stores 2 to the N single on/off bits. To consider the RAM as a group of 2 to the N neurons would be misleading, although it could be thought of as one neuron having a single combined output. It is tempting to regard the discriminator as being a Hubert and Wiesel hypercolumn, or a Marr Super Hypercomplex group. That is, a line detector, a movement detector or other special unit. In fact analogies of this sort with the other networks also make sense.

96. General Comparisons. It is not reasonable to assume that the brain is one large Hopfield or any other sort of net. Combinations of networks are more likely, and are briefly discussed in section 9. It has been shown that each of the networks has an effective data capacity within its limitations, and I do not propose to deal here with the exact limits, as there are many variables affecting the limitations. Dobson in [4] deals thoroughly with some of the comparisons but I have considered the following:

97. Performance. In terms of speed in computer simulation, Dobson's nets win hands down, with the WISARD net second, and Hopfield's net last. The speed in hardware simulation can only be estimated.

98. Accuracy of Recall. The data accuracy comparison is interesting because each net deals with the data in different ways. In terms of accuracy of recall, the Dobson net is best. In terms of autoassociative recall the Hopfield net is most useful as it will return the nearest likely image, the Dobson net may return nothing , but the WISARD net would at least tell how near the pattern is to one previously learnt.

99. Object Recognition. Categorisation and generalisation, that is recognising groups of similar stimuli, is important in evolutionary terms. The adaptable and versatile creature is more likely to survive a

capricious environment if it has this skill as the recognition of classes or groups of dangerous situations is important. This adaptability is the subject of study in Intelligent systems. S.Smith (Carnegie-Mellon Univ) has developed some "genetic algorithms" that illustrate the benefits of optimisation. Certainly the Hopfield and WISARD nets are ideal for this function.

100. Functionality. The usefulness of the systems outside the neurological perspective has been proved for the WISARD system. No doubt the other networks would benefit from a hardware (i.e. rapid!) implementation which would then see their use in practical situations.

---------- 0 ----------

SECTION 9 - FURTHER PROJECTS AND DEVELOPMENTS BY COMBINING NETWORKS IN A COLLABORATIVE MODE AND USING FEEDBACK

101. Some speculations about higher level structures have already been touched upon. Aleksander in [7] covers some of the aspects of using feedback to resolve low confidence decisions about certain patterns. This is done by having the histogram display as part of the pattern learnt. As the system moves nearer the correct solution the histogram becomes nearer the "correct" one, so causing a stronger response etc. Feedback arrangements have already been tried to give sequence dependent patterns, eg. pattern A associated with pattern B, B with C, and say C with D. The results of closing the loop D with A, and then giving random input may produce dream-like sequences (see Section 11).

102. The following arrangement is to be followed up after this dissertation. The memory state for a particular class can be loaded in and out of a WISARD system. If the memory state is put into an autoassociative network e.g Hopfield or Dobson type, and several classes of object are remembered in this way, the overall system could work as follows:

- Teach and store the classes.
- Present a test pattern, P, and fill the RAM memories as would be done for teaching.
- Give the autoassociative memory the RAM states S.
- Put the output of the autoassociative memory, O, into the RAM network as their new state.
- Present the test pattern, P, again, but this time get a response.
- If the confidence is "inadequate", THEN teach the RAMs with the test pattern for a longer time, and return to c) ELSE output the class represented by the output O.

This arrangement would deal with only one object at a time in the input array. There are possibilities for developing networks of "feature detectors" on this sort of basis. One of the main problems with this sort of experiment will be its slowness, without some parallel architecture being included. This is discussed in the next section.

---------- 0 ----------

SECTION 10 - THE PROBLEMS OF CONSTRUCTING PARALLEL HARDWARE FOR NETWORK SIMULATION

103. Parallel architecture discussions recently published [40,41], have covered the problems for solid-state architecture in terms of packaging, power supply, heat removal, connectivity, input and output interfacing, interference and reliability. For network simulations all of these are important, but particularly packaging and connectivity. Why?

104. The problem for a network is that it is parallel, and requires a large number of inter-connections. In a minimum useful sized network, with a 1000 neurons, the net needs one million connections. Squeezing these in, reliably, without interference seems an impossible task. So how does the brain manage it? Some neurons receive several thousand connections. There are five important differences as follows:

- The neuron uses its power in a manner which generates absolutely minimal heat.
- The neuron is immersed in a nutrient bath, that is: it is self powered.
- Its connections branch only when necessary, and can take the most direct route, ie. it is nearly irrelevant what other parts of the brain it goes through.
- The neuron does not function serially at any level. Within the axon there is n-way chemical transport as well as action potentials. Each of these functions does not need a separate wire.
- Because of that, an axon or dendrite can branch only where it is absolutely necessary.

105. It can clearly be seen that the brains multi-level communication modes allows far more "information" to be dealt with through any one "wire", as well as needing less "wires" because there is no need for power supply etc.

106. There have been many good ideas developed by "taking a lesson from nature" (so-called biomimetics). It may be that this is not such an old cliché after all. In [39], Bond discusses what is required "from the biologists" to give the programmers a tight enough specification to produce conventional computer programs in the information processing vein. I think that such an approach needs to look far harder at the benefits of thinking in network terms.

---------- 0 ----------

SECTION 11 - IMPLICATIONS OF THE SIMULATIONS FOR UNDERSTANDING COGNITIVE PROCESSES

107. I do not think that the network theories discussed here have much to offer to the understanding of cognitive processes, any more than studying Physics explains a computer program. The problem is one of logical typing. Human consciousness is a mechanism which allows us to read "chunked" bits of brain state. We are not aware of what a particular neuron is doing. Hofstadter [25] talks about intelligence and says:

"Intelligence depends crucially on the ability to create high level descriptions of patterns and organisations and then think on that level, outside the system, about the system" [my emphasis]

and again:

"Intelligence shows that it has a power of generalisation, a feeling for form, and a capacity for rapid modification of hypothesis"

These passages illustrate that trying to talk about cognition and neurons in the same breath is sure to lead to trouble.

108. Just to make my position clear, I feel these networks do not explain cognition, BUT... the networks can be discussed as hypothesis, and modified of course. They obviously allow us to make some interesting suppositions about memory, as has been discussed in the previous sections, and about vision and other sensory input.

109. Crick [9], has linked the benefit of the "unlearning", that can be applied to Hopfields net, to dream sleep. In [8], Aleksander provides one critique of this theory. The theory is interesting as far as it goes. Aleksander puts forward an alternative theory; that the dream is perceived - as the net (asleep and so without external input to force it along the trajectories in state space, and under random-like PGO input) re-traverses the trajectories passed along during the day and so returns to the neutral regions. As Aleksander concludes :

"In the end, if pushed hard, I must admit that both theories are somewhat arbitrary and coexistent. It may be an interesting task for the psychologist to devise a test which would differentiate between the two theories, or possibly even deny them both."

110. As our networks become more sophisticated, the theories that the networks provoke will become more sophisticated too. Aleksander, in a seminar, proposed Six Laws of Intelligence, that (paraphrasing his deep discourse), you have intelligence when you:

1) are self-aware, and aware that you are self-aware;
2) able to sense the world and other beings and perceive that they are self-aware;
3) can appreciate that they have different motivations and views of the world to yourself;
4) can conceive of what their view(s) of the world may be;
5) can reason from those points of view and synthesise them with your own ...
6) and lastly be able to act, interact, and effect change in the world in line with those things - anticipating, adapting and changing over time - and so changing the nature of your intelligence in line with the real-world context.

Such a definition would only apply to 'Strong AI' or 'General AI'. The other various kinds of simulations and emulations of ‘intelligent’ behaviour succeed as far as they do because of the human ability to anthropomorphise and attribute intelligence where it does not exist.

---------- 0 ----------

SECTION 12 - CONCLUSIONS AND THOUGHTS ON FUTURE WORK

111. In this dissertation I have developed the thesis that there is a lot to be gained by a better understanding of network theories. There certainly is a lot more research in these areas, from many points of view, genetics, computation, psychology, intelligent systems, sociology, cellular automata and many others. The fact that networks have "emergent properties" is a fact of life, of which we have probably not taken enough notice.

112. Even if the emergent properties of the networks implemented here have little to offer, at this stage, to the understanding of cognition in the brain, it is still reasonable to consider the brain as "units" of networks, working (relatively) autonomously, and combining their (continuous) outputs in cognition.

113. It may also be relevant to consider human intelligence not just as an individual property, but also of society, culture, upbringing and genetic history. It may be that there is a "network" of interactions at the cognitive, social and cultural level that has more interesting "emergent" properties than small associative memory network simulations.

114. There is a great deal of interesting research and study to do in this area, and I hope to develop the line of connected networks, such as the one I mentioned in section 9.

---------- 0 ----------

ACKNOWLEDGEMENTS

I would like to thank the following people for their help and encouragement:

My supervisors and lecturers Dr M.Wright, Dr A.Johnston, Dr M.Elstob and Dr V.Dobson.

Professor Igor Aleksander who provided many inspiring ideas and valuable criticisms.

Professor Liam Hudson who allowed me to access private communications with F.H.Crick, and passed on to me a program on disc from him which was an explanation of the Hopfield network with which Crick has been working.

The technical staff in the Psychology Department at Brunel University.

Judy Beautement for all her hard work in the background.

The Manpower Services Commission for providing a grant for the year.

---------- 0 ----------

REFERENCES AND BIBLIOGRAPHY

	Title:	Author:	Publication:
1.	A critique of the Hopfield-Crick theory of dreams and unlearning.	I. Aleksander	Private Communication.
2.	Emergent intelligent properties of pattern recognition nets.	I. Aleksander	Pat.Rec.Let. 1983.
3.	Memory networks for practical vision systems.	I. Aleksander	In Physical and Biological processing of images. Ed. O.J.Braddick & A.C.Sleigh.
4.	Learning in marine snails (hermisenda).	D.L. Alkon	Sci.Am. 83.
5.	Parallel models of associative memory.	Eds G.E. Hinton & J.A. Anderson	Lawr.Elrl.Assoc.
6.	Microelectronic packaging.	A.J. Blodgett	Sci.Am. July 83.
7.	Artificial Intelligence and Natural Man.	M.A. Boden	Harvester Press. (O.U. book).
8.	The development of an analogue spreading activation model for schema matching in semantic memory.	W.R.K. Bradbeer	New Psy. 1984.
9.	Psychology and biology of language and thought (essays).	Ed. E. Brown	Academic Press.
10.	Essays on cellular automata.	Ed A.W. Burks	Un.Illinois.Pr.
11.	Seminar.	Prof. Fergus Campbell	Brunel University. June 84.
12.	Determinants of random stability.	J. Ford, F. Vivalda, W.M. Visscher & G. Casati	Phys.Rev.Lett. 52,(1861) 1984.
13.	Are all inhibitory nets universal synthetical computers?	V.G. Dobson	Inst.of Cyb. Monograph 23. 1977.

14.	Differences in performance of associative nets using correlative, additive and subtractive modification rules.	V.G. Dobson	Brunel University. 1984.
15.	Solid-state superlattices.	G.H. Dohler	Sci.Am. Nov. 1983.
16.	A mathematical model for synaptic efficacy.	J-P. Changeux & J.L. Gouze	Biological Cybernetics. 46(3) 1983.
17.	Associative memory models.	D.J. Willshaw & H.C. Longuet-Higgins	Ed. B.Meltzer & D.Michie. in Machine Intelligence Vol 5 1970. Edin. Univ.Press.
18.	Godel, Escher, Bach.	D.R. Hofstader	Penguin.
19.	Introduction to occurrence systems.	Anatol.W. Holt	-
20.	Neural networks and physical systems with emergent collective computational properties.	J.J. Hopfield	Proc. Natl. Acad.Sci.USA. Vol 79. 1982.
21.	Unlearning has a stabilising effect in collective memories.	J.J. Hopfield	Nature Vol 304. p158. July 83.
22.	Metabolic stability and epigenesis in randomly constructed genetic nets.	S.A. Kauffman	J.Theo.Bio. 22. 1969.
23.	Associative Memory.	T. Kohonen	Springer Verlag.
24.	The working brain.	A.R. Luria	Penguin Mod.Psy.
25.	Conscious and unconscious perception.	A.J. Marcel	Cog.Psy 1983.
26.	Seminar.	A.J. Marcel	Brunel University. May 84.
27.	Vision.	D. Marr	Freeman.
28.	The function of dream sleep.	F.H. Crick G.	Nature Vol 304. July 83.

29.	Theory of self reproducing automata.	Von Neumann	Ed. A.W.Burks. Univ.Illinois.Pr.
30.	Principles of Artificial Intelligence.	N.J. Nilsson	-
31.	Neural assemblies (an alternative approach to A.I.).	G. Palm	(Studies in brain function, Vol 7) 1982.
32.	A model for knowledge and its application to discourse analysis.	B. Phillips	Am.J.Comp.Ling. 1978.
33.	Schemata, the building blocks of cognition.	D.E. Rumelhart	Theoret.Issues. 1980.
34.	Principles of neural science.	E.R. Kandel & J.H. Schwartz	-
35.	Minds, brains and programs.	J.R. Searle	Behaviour and Brain Sci. 3. 1980.
36.	Physical and Biological Processing of Images.	Eds. O.J. Braddick & A.C. Sleigh.	Springer Verlag.
37.	Introduction to automata theory, languages and computation.	J.E. Hopcroft & J.D. Ullman	Addison Wesley.
38.	The Brain.	Ed. Various	Sci.Am books.
39.	What does it mean to understand language?	T. Winograd	Cog.Sci 4. 1980.
40.	Artificial Intelligence.	P.H. Winston	Addison Wesley.
41.	Autopoiesis, a theory of living organisation.	Ed M. Zeleny	-

---------- 0 ----------

APPENDICES

A. NETCHAIN program. Listing and sample output.

B. LAYOUT and MAKERAND programs. Listing and sample output.

C. RAMSTATES program. Listing and sample output.

D. HOPNET18, listing.

E. HOPNETPB, listing.

F. HOPDEMPB, listing.

G. DOBNET18, listing.

H. DOBNETPB, listing.

I. MINIWSRD, listing.

[N.B. All the above programs are explained in sections 4 - 7]

J. Training sets for the associative network (section 5) and the subtractive network (section 6).

K. Training sets for the WISARD network (section 7).

l. Two floppy discs with all the above programs, plus boot tracks etc.

---------- End of P. Beautement's MSC thesis main text ----------