The attraction of parallel distributed processing for modelling cognition

entation and processing of information in connectionist networks is
l. Decisions are reached by consensus of a large number of simple com-
taking place in parallel as stimulus information interacts with stored
. In consequence, connectionist memories display many human charac-
hey are relatively immune to damaged components within the system or to
t; they allow retrieval by content; they are likely to retrieve typical
rom categories.

ecade has seen an explosive growth in the connectionist modelling of
rocesses, with simulation of most of the classical experimental paradigms
e psychology. One reason for this enthusiasm is that, independent of their
modelling human performance at any particular cognitive task, all con-
models exhibit some general characteristics which are shown by human
rocesses and distinguish them from non-biological computational systems
mputer programs: They still perform reasonably well after minor damage
ents of the system; they still perform reasonably well if their input is noisy
te; they allow memory retrieval by content.

hapter we will look at two aspects of connectionist systems which are
e for these characteristics. Like the principles of interneuronal com-
n described in chapter 1, these are based on general observations of brain
First, knowledge representation is *distributed* across many processing
ond, computations take place in *parallel* across these distributed
ions. The result is that conclusions are reached on the basis of a consensus
lculations rather than depending on any particular one.

rinciples put connectionist models in direct contrast to many traditional
cognitive psychology or artificial intelligence where knowledge representa-
l and computation is serial. In general, such models are not immune to
damage or resistant to noisy input. So a traditional model of, say, s
reasoning, might give as good a fit to the experimental data as a connection
but it would do so without exhibiting the full range of human characteri
performed the task.

The representation of knowledge in connectionist net is distributed

Traditional models of cognitive processing usually assume a local represer
knowledge. That is, knowledge about different things is stored in
independent locations. In a traditional model of reading aloud, for
information about how to pronounce the letter string DOG is stored in one
information about how to pronounce the string CAT in another. What
more natural? The two pieces of information are independent and would be
at different times. So storing them independently makes obvious se
information storage systems we are familiar with in everyday life—dic
telephone directories, computer discs—use local representation. Each disc
of information is stored separately. How else could it be done?

In connectionist models information storage is not local, it is *distributed*
no one place where a particular piece of knowledge can be located. Cor
segment of network at the bottom of figure 1.2 in chapter 1, part of a large
which is learning to read aloud. Any input, such as the letter string DO
excite units and connections all over the network. Learning takes place by
the weights of the connections leading to all output units which have an
level of activity. The knowledge of how to pronounce the input DOG is d
across many different connections in different parts of the system. It is the
effect of all these connections which produces the pronunciation, not any
of them.

The concept of distributed storage may be difficult to grasp at first bec
counter to our everyday experience of information storage systems. The co
which contain the system's knowledge about how to pronounce DOG are
as those with the knowledge about how to pronounce any other letter strin
knowledge that the network contains is superimposed on the *san*
connections. Intuitively this may seem entirely implausible. How can the sa
weights store independent and even contradictory pieces of information? A
see, it can be done, and some of the emergent properties of such sy
intriguingly similar to properties of human cognitive processes. But for th
this will have to be taken on trust. There are no familiar information storag
which use distributed coding, so analogy to a familiar system is not possibl

...uted representations are damage resistant and fault ...t

...considers the structure of the brain it is remarkable that it ever manages to ...he correct conclusion about anything. By any conventional standards ...e an entirely unsuitable medium for computation: they die throughout the ..., causing random loss of stored information; they have a finite probability ...ven when they are not engaged in signal processing; the response of a ...any particular input is probabilistic, not fixed.

...ok at the firing pattern of a single neuron the problem that probabilistic ...cause for the system will become clear. The upper part of figure 2.1 shows ...e response of a single neuron in the visual cortex to a stimulus presented to ...he stimulus is presented at time 0. Time after the presentation of the ... shown on the horizontal axis. The vertical axis shows the neuron's firing ...s an average background firing rate of a few spikes per second. When the ... presented a signal is superimposed on this. About 50 ms after the stimulus ...e neuron fires strongly for about 30 ms. 100 ms later it fires again, rather ...ely, for about 50 ms. This is the signal that the neuron transmits to the ...ons to which it is connected, indicating what pattern of stimulation it has ...

...ms fairly straightforward. But the histogram was obtained by summing the ...erns over a number of presentations of the stimulus. The lower part of the ...ws 12 different occasions on which the same stimulus was presented. Each ...euron fires there is a vertical spike. If we look at these individual trials the ...ich emerges is much less clear than that suggested by the overall average at ...On trial 5 the initial burst was missing. On trial 10 the second burst was ...On trial 11 the neuron did not respond at all. It is clear that the 'signal' ... histogram at the top shows is an idealised average. On any given trial the ...ll only approximate this, sometimes quite closely, sometimes not at all. ...the system produce a reliable response on every trial when the individual ...ts only produce their signal on average across trials?

...ocessing components in a conventional digital computer produced random ...us output, a different response to the same stimulus on different occasions ...red from random component drop-out, the system would be totally un-...e! Sometimes it would work correctly, but if a computation required access ...tents of a missing memory unit, or a burst of noise obliterated a signal, the ...uld be garbage. Although the components in the brain can fire and die at ...he computations performed by the brain are not unpredictable. With minor ... becomes a little slower and less accurate, but it still produces roughly the ...ver. It has to suffer serious damage before it produces nonsense.

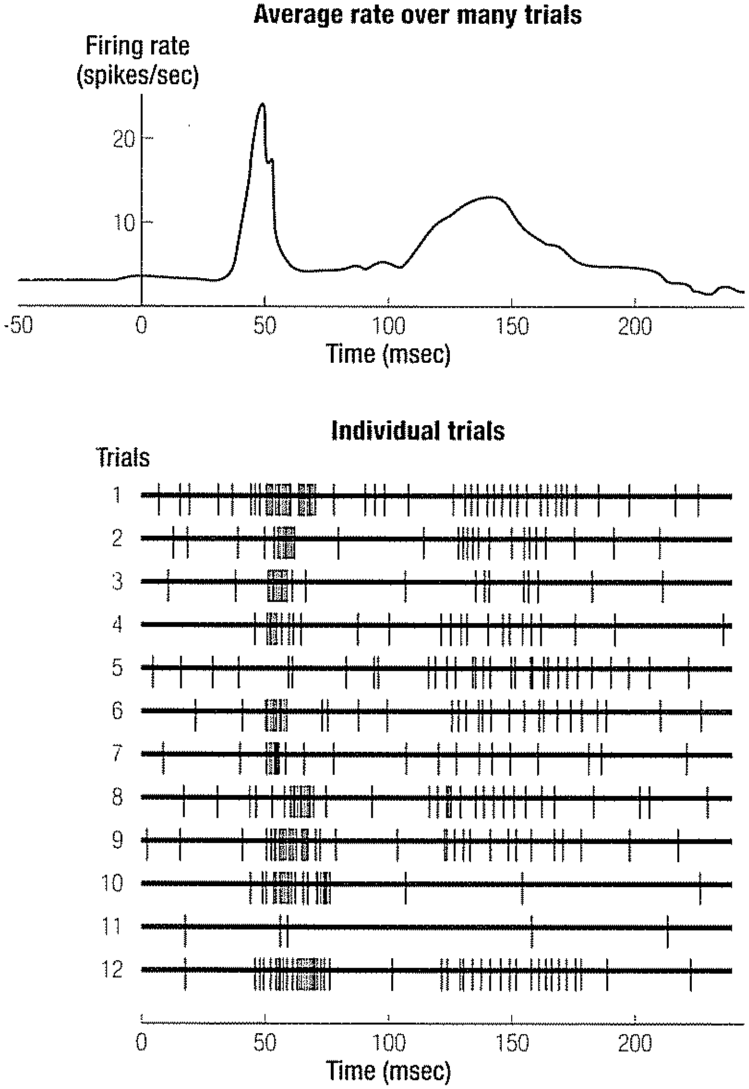


Figure 2.1 The response of a neuron in the visual cortex for 250 ms after stimulus presentat... The firing rate of the neuron summed over a number of presentations of the stimulus. Lower: ... on 12 of the trials which contributed to the average response shown at the top. Each v... represents the time at which the neuron fired. (Based on Morrell 1972.)

The brain escapes the consequences of the unpredictable behaviour of i... neurons because its computations are performed in parallel on representatio... are distributed over many neurons. No one neuron plays a crucial role in pr... The overall result is the outcome of many distributed sub-computations... individual components of the calculation are not accurate, the ensemble a...

theless give an answer which is accurate enough. When the memory equired for a calculation in a localist information storage system is the result is disaster. If the first page of a dictionary is missing, there is no ecking whether *aardvark* is really spelt like that. But, in a connectionist ere is no such thing as 'the memory location required for a calculation'. on and calculation are spread across the network. If one unit or connection work is damaged, others can make up for the missing part.

tem will, of course, be slightly less accurate if a connection is lost. But the f loss is quite different in localist and distributed systems. Damage to a stem causes some information to be lost totally while other information is d. In a distributed system any damage causes partial loss of a range of on. As damage increases, the performance of the system inevitably begins to a small amount of damage may have no noticeable effect on the output of n. The ability of brains and connectionist models to continue to produce a e approximation to the correct answer following damage, rather than ng catastrophic failure, is an example of fault tolerance referred to as *legradation*.

ctionist networks allow memory access by content

nemory is content-addressable. That is, you can access a memory by using t of the information contained in the memory (the *content*) as a retrieval is unlike retrieval from familiar forms of information storage, such as ies, telephone directories or computer discs. In these, the place where the ion is stored has an *address*. The only way to access information for s with the address. For example, in a dictionary the address is the spelling of ; the information stored with this address is the definition of the word. In a system a typical address is the name of a file; the information which can be with this address is the contents of the file.

derstand the difference between accessing a memory by address and by contrast retrieving information from a dictionary with obtaining the same ion from a person. Imagine that you want to know the name of a man-made lt across a valley, to contain water to build up a head of pressure to generate y. With a dictionary, there is no way to access the location where this ion is stored and extract the missing piece, the word 'dam'. If you start with ess (DAM) you can access all the information stored at the address. Without n access nothing. In contrast, a person given part of the information would be able to retrieve the rest. Unlike a dictionary, human memory allows a any part of the information that forms the memory. One of the reasons nectionist models of human memory are attractive is that content addressability follows as a natural consequence of their distributed structure. addressability can be built into localist storage but only by adding a compl referencing system.

Any information processing system which works in the brain must be fault because the signals it has to work with are seldom perfect. There is a random ent to neuronal firing; speech is usually heard against a background of othe objects rarely present the same image on different occasions. An attractive content-addressable memory is that it is inherently fault tolerant. Imagine asked you to guess who they were thinking of: 'This man was a British Con politician. He became Prime Minister in 1978 and was Prime Minister du Falklands War. He was ousted from office by his own party, being held res for the fiasco of the Poll Tax. He was eventually replaced as Prime Minister Major.' You could probably suggest 'Margaret Thatcher' as an answer de fact that some of the information is incorrect. Mrs Thatcher did not becon Minister until 1979, of course. With content-addressable memory the w evidence pointing to one answer can overcome other evidence that is incons best fit solution can be chosen even if it is not perfect. This is unlike memory in which access by address is the only possibility. Any error in the address wi failure. A search of *Who's Who* using the address 'Margaret Patcher' would nothing, despite the fact that most of the search term fits an existing entry.

Retrieving information from a distributed database

To see how a distributed system with parallel processing works in practic look at retrieval from a simple connectionist memory described by Mc (1981).[1] This memory demonstrates content addressability and fault tole also shows typicality effects in retrieval—if asked to retrieve a random men category it will produce a typical member. Considering the simplicit simulation it demonstrates a remarkable range of human characteristics in retrieval.

Imagine that you live in a neighbourhood where many of your male acqua belong to one of the two rival local gangs, the Jets or the Sharks. Your kn about these characters will come from a succession of independent episo night Fred emerges from behind a bush and offers you some white powder. Dave and his wife trading insults as she drives off with a car full of s Everyone in the bar is laughing because Don has been admitted to colleg basis of forged examination results. Nick hangs out with Karl whom you kn a Shark. All these pieces of information about your neighbours are g

[1] The Jets and Sharks memory system is not implemented in tlearn but its properties and th described in the text can be explored using the iac program in McClelland and Rumelhart (198

		Contents		
Gang	Age	Education	Marital Status	Occupation
Jets	30s	JH	Married	Burglar
Jets	40s	JH	Single	Pusher
Jets	40s	JH	Single	Bookie
Sharks	30s	HS	Divorced	Pusher
Sharks	30s	Col	Married	Burlar
Jes	30s	HS	Single	Bookie
Sharks	40s	HS	Married	Burglar
Jets	20s	HS	Single	Pusher
Jets	20s	Col	Single	Pusher
Jets	20s	JH	Divorced	Burglar
Jets	20s	HS	Married	Pusher
Sharks	30s	JH	Single	Bookie
Jets	20s	JH	Divorced	Burglar
Jets	20s	JH	Married	Burglar
Sharks	40s	HS	Married	Bookie
Sharks	20s	HS	Single	Burglar
Jets	20s	JH	Married	Burglar
Jets	30s	JH	Single	Bookie
Sharks	30s	HS	Single	Bookie
Sharks	30s	Col	Married	Bookie
Sharks	30s	HS	Single	Pusher
Sharks	30s	Col	Married	Pusher
Jets	20s	HS	Single	Bookie
Sharks	30s	Col	Married	Pusher
Jets	30s	JH	Single	Pusher
Sharks	30s	HS	Divorced	Burglar
Jets	20s	Col	Married	Bookie

McClelland 1981.)

ed over the years. Fred is a pusher; Dave is divorced; Don went to college; Shark.

2.1 shows how this information might be stored in a conventional tion storage system. The system has a set of storage locations, corresponding of cards in an index file or a set of files on disc in a computer, for example, ded by an address. The address is the name of the person. Each new piece of tion relating to him is stored at the location headed by this address. In this imulation we imagine that we have information about the **Gang** each person to (*Jets* or *Sharks*), a rough idea of his **Age** (*20s*, *30s* or *40s*), the extent of his on (*Junior High*, *High School* or *College*), his **Marital Status** (*Married*, *Single* rced), and his **Occupation** (*Bookie*, *Burglar* or *Pusher*). The format used in

table 2.1, address + contents, is a logical way of storing the information name *Alan* is the key which binds one set of information together, *Clyde* another set, and so on.

This form of storage is efficient for retrieving information in response to like 'Is Fred a pusher?'. The question contains the address, and the addr directly to the place where the information which provides the answer is sto it is not so good for answering other enquiries. If you are asked 'Do you k name of a pusher?', the only way is to search through the list of addresses find one where the information *Pusher* is stored under **Occupation**. Altho information *Pusher* is stored at many locations, it does not form part of the So an answer to this question cannot be extracted directly from the memory.

Admittedly, with this particular database it would not take long to find a you searched addresses at random. But a more realistic representation of kn of these people would include many unique pieces of information, such as that Fred's grandparents came from Ballylickey. The only way to store system like that shown in table 2.1 is as the fact 'grandparents came from Bal at the storage location with the address *Fred*. The question 'Whose grand came from Ballylickey?' could only be answered by random search of the a until the one containing that information was found. This might take a lo although you would get there in the end. But a human memory would not like that. If you could remember the information at all you would usually the answer reasonably quickly. This is because human memory can be acc *content*—any part of the knowledge base can be used to access any oth 'Ballylickey' can be used as a cue, and will lead to *Fred*. The information system shown in table 2.1 is perfectly logical. Indeed, it is probably the sort o you would use if you were asked to store the information about the Jets and But human memory cannot be organised like this. A memory organised like does not allow content-addressable retrieval; human memory does.

Another way of seeing why human memory cannot be organised so that only possible by address is to consider how you would answer the question ' the Jets like?'. Table 2.1 allows easy access to information about individual it offers no simple way to answer questions requiring generalisations number of entries. Human memory does allow generalisations across memory. A person who knew these two gangs could probably tell you that were younger than the Sharks without having to think very hard. Altho method of information storage shown in table 2.1 seems natural, it canno way that human memory is organised.

(1) Setting up a distributed database for the Jets and Sharks base. Mc explored the consequences of storing the information about the Jets and S a distributed system. The architecture of his system is shown in the upper

. To store the information in table 2.1 we need to represent facts about ang Membership, Age, Education, Marital Status and Occupation. Within ese areas of knowledge there is a node corresponding to the possible values one could have. So in the **Age** area there are nodes for *20s*, *30s* and *40s*, in on for *Burglar*, *Bookie* and *Pusher*, in **Gang** for *Jet* and *Shark*, and so on. lel might seem to be localist rather than distributed because there are l nodes to represent specific concepts. But, as we shall see, the underlying of the model are parallel and distributed. The result of any input is ed by interaction across the entire database. The localist coding of concepts make it easy to see what the model is doing in the examples which follow.

ory is formed by setting up a link between two nodes. If we discover that bookie we set up a positive connection between the *Sam* node in the **Name** the *Bookie* node in the **Occupation** area. If we then discover that he is ve set up a positive connection between *Sam* in the **Name** area and *Married* l **Status**, and between *Bookie* and *Married* (since we now know of a bookie arried). To store all the information we know about Sam we would set up inks between all the possible pairwise combinations of *Sam* in **Name**, *Jet* in)s in **Age**, *College* in **Education**, *Married* in **Marital Status** and *Bookie* in on. The way that McClelland did this is shown in the middle of figure 2.2. a Person node in the central region of the model and then made a positive on between this and each fact that was related to that person. The result is o setting up all 15 links necessary to represent these facts individually, res fewer links. Person nodes were then set up for each of the people in table the necessary links formed to represent everything that is known about

ottom part of figure 2.2 shows the connections between the various nodes up as information about five of the people is added to the system. This also second element of the model. There are mutually inhibitory connections each of the exemplar nodes within a knowledge area. There must be some y links in a network like this where everything is interconnected or the result ting any node would be that everything in the network would eventually maximum activity level and no differential response to different inputs possible. These connections represent the fact that, for example, if someone 20s, he cannot be in either his 30s or his 40s. This fact could have been im- ed by making negative connections from each person node to all the things t. Building mutually inhibitory links between alternative instance nodes

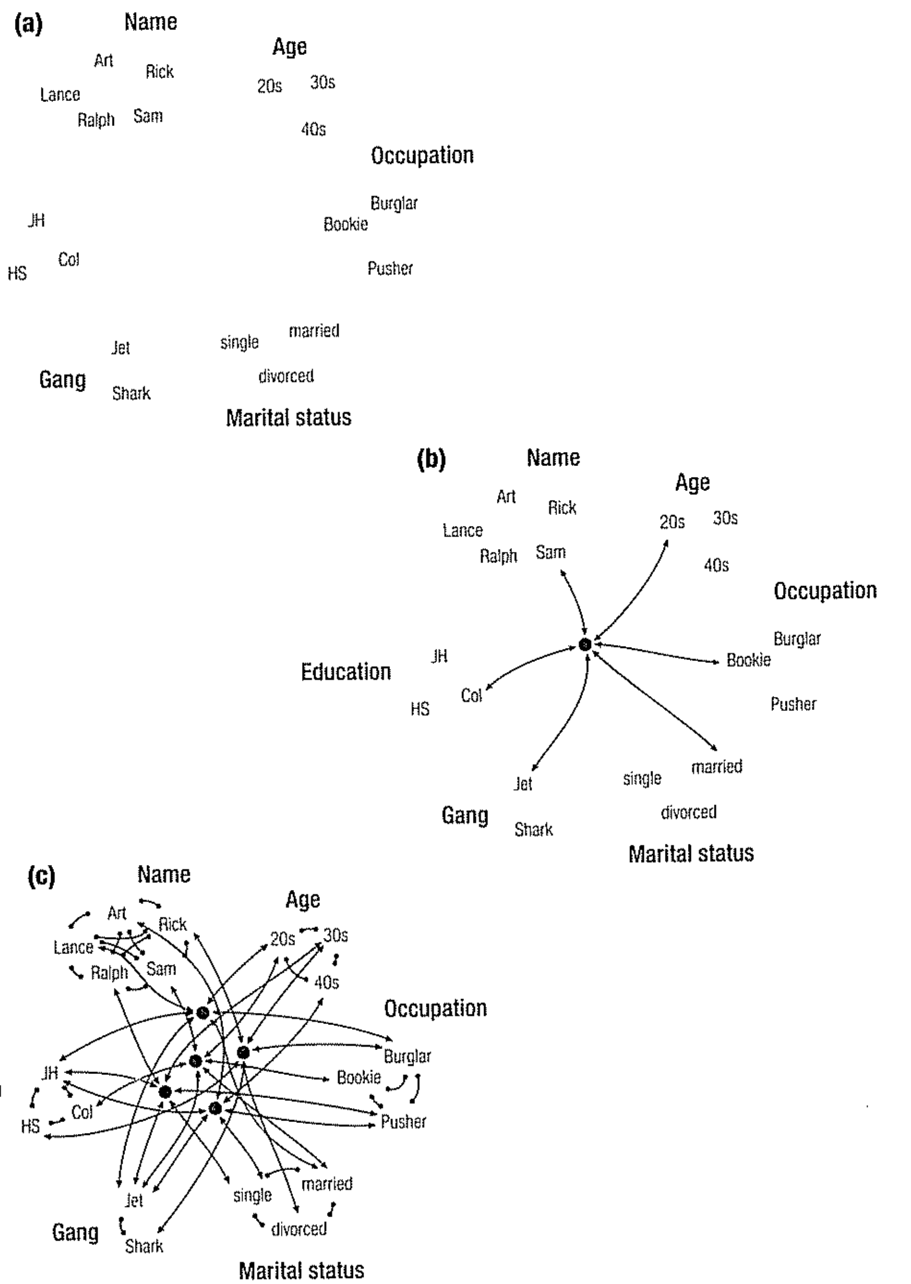
eem that the Person node corresponds to the address for the information just as the name Sam le 2.1. When information is retrieved from the net the Person node cannot be accessed, so it used as a retrieval address. It is just a convenience which reduces the number of connections set up the model and makes the operation of the model easier to follow.

within an area has a similar effect on the performance of the net but greatl the number of connections required. As we shall see, the way that inhibitio into this network has an important role to play in the way the model runs.

(2) Running the network. The nodes are the processing units of the mo act like the one at the bottom of figure 1.3 which was described in chapte has an activity level associated with it. When the model runs, each no activity to all the other nodes to which it is connected, in the way des equation 1.1. That is, the input to the receiving node is the product of th level of the sending node and the weight of the connection between the simple model the weight of all positive connections between nodes is +1 ar negative connections is -1.[3] So, when the model runs, every node wh positive activity level tries to increase the activity level of every node to whic positive connection, and to reduce the activity of every node to which negative connection. The net input of each node is determined by summ negative and positive inputs (as described in equation 1.2). The net inpu converted by an activation function to an activity level. The exact for activation function used by McClelland was not the same as any of those figure 1.4 but had the same effect as the sigmoid function in figure 1.4(c) o the maximum value which the activity level could reach, and of slowing the activity level with changes in net input as the unit's activity level appro maximum value. In a single processing cycle the activity level of eac computed by summing its inputs and converting these to an activity level activation function. On the next cycle these new activity levels are used to the new net inputs to each unit, and thus their new activity level. This is c until the net reaches a steady state. That is, until each node in the network constant activity level.

(3) Retrieval from the database. To test the memory performance of this s ask it a question such as 'Can you remember the name of a pusher?'. This i activating the *Pusher* node in **Occupation** and waiting to see which unit active in the **Name** area as activity passes round the network. The activ nodes starts at a level of -0.1. Activity of the *Pusher* node increases the acti of all nodes to which it has a positive connection and decreases the activ those to which it has a negative connection. Once the activity level of a n

[3]There is no gradual learning phase in Jets and Sharks. Facts are given to the model complete the weights do not develop as knowledge is acquired as they would in a conventional co model. The way that information is entered into this system is an example of Hebbian lear will be discussed in detail in chapter 3). If two things are mutually consistent (e.g. being in yo being a burglar) a positive connection (via the appropriate Person node) is made between t things are mutually inconsistent (e.g. being in your 20s and 30s) a negative connection is ma them.



it excites all the nodes to which it has a positive connection and inhibits all which it has a negative connection. Eventually the system reaches a steady which the activity level of each node is constant, either because it has reached mum or minimum permitted value, or because its negative and positive

Figure 2.2 The architecture of McClelland's system for storing the information about t Sharks shown in table 2.1. (a) Each cloud represents an area of knowledge about the member gangs, with the nodes within a cloud representing possible instances (b) The information a represented by setting up excitatory connections between the facts that are known about done by setting up a Person node (the black node in the central circle) and linking this to all nodes which represent his properties If any one of these nodes becomes active these links will the nodes representing his other characteristics will be activated. (c) The excitatory connectior to represent all the information about five members of the gang have been entered connections (links with filled circles on their ends) have been set up between competing inst within each area of knowledge. When the model runs, any node which has a positive activi inhibit any other node to which it is connected by one of these links. (Based on McClelland 1

inputs are exactly balanced.[4] The **Name** node which is most strongly activa steady state is reached is the system's answer to the question.

Does such a system behave like human memory? Apart from being able t the information it had been given directly, by answering such questions does Fred do?', anyone who knew these people would find it easy t questions like 'Do you know the name of a pusher?' or 'What are the J These are the sort of questions which it is difficult to answer with a localis store organised like table 2.1. Will this distributed, connectionist memory sy them any easier?

With the aid of the bottom part of figure 2.2 it is possible to get some ide happens when the system runs. To see what answer the system will retrieve asked: 'Can you remember the name of a pusher?' the *Pusher* node is activa activity passes along all the connections from the *Pusher* node. So the *Ralp* **Person** nodes become excited because there is a positive connection to th *Pusher*. (In the real model all the **Person** nodes of pushers would be exc simplicity we will just follow two of them.) The *Bookie* and *Burglar* node inhibited because there are negative connections to them from *Pusher*. In processing cycle the *Ralph* **Name**, *Jet*, *30s*, *JH*, *Single* and *Pusher* nodes a by the *Ralph* **Person** node, and the *Art* **Name**, *Jet*, *40s*, *JH*, *Single* and *Pus* become excited by the *Art* **Person** node. At each succeeding cycle every nod active influences every node to which it is connected by an amount which de its activity level and in a direction which depends on whether the connectio them is excitatory or inhibitory. So, for example, the fact that the *30s* node by the *Ralph* **Person** node will in turn cause excitation of all the **Pers** connected to *30s*, and inhibition of the *20s* and *40s* **Age** nodes. At the same excitation of the *40s* **Age** node by the *Art* **Person** node may be sufficient to

[4]In McClelland's model the activity of each unit also decays on each cycle by an amount pro its activity level. This affects the dynamic behaviour of the net but to understand why the n steady state it can be considered as another negative input contributing to the balance betw and negative inputs to each unit.

use excitation of all **Person** nodes connected to *40s* and the inhibition of d *20s* **Age** nodes. After several processing cycles the activity level of every e system is being influenced by a mixture of positive and negative inputs. it soon becomes impossible to keep track of the patterns of excitation and and predict whether the system will reach a stable state, and if so, what ited and what depressed. The only way to find out what the system will do e to stimulation of any of its nodes is to run a computer simulation of the em.

d now be clear that a connectionist memory system is totally unlike a nal memory such as a computer filing system. In a conventional system ent pieces of information are stored separately. When a specific piece of on is accessed, it and it alone is retrieved. But in a distributed connectionist n attempt to extract any information from the system leads to a flow of and inhibition throughout the system to everything which has any relation ormation. This results in many different nodes becoming active. What is is the information which corresponds to the most active node(s) once this stabilised. If one is accustomed to information retrieval from a con- , non-connectionist system such as a telephone directory, this might seem If you looked up Tom Brown's number in a connectionist telephone the number retrieved would be influenced by the entries of everyone with a ame or a similar number to Tom's. This would not be useful. You do not number retrieved to be influenced by the fact that there happens to be called Tim Brown who has a telephone number quite unlike Tom's. You information stored at the location with the address 'Tom Brown' and else. But the interference which a connectionist system allows during between related items of stored information turns out to have some g and useful properties.

ntent-addressable memory in Jets and Sharks. To see whether this memory lows access by content, we can ask it the question 'Do you know the name her?'. To do this we activate the *Pusher* node, leave it on, and see which des becomes activated. Figure 2.3 shows the activity level of three of the des as a function of the number of processing cycles for which the system allowed to run. All the pushers names initially become activated. Most of e *Oliver*, quickly return to their resting level. But *Fred* and *Nick* both ncreasingly activated. After about 50 cycles *Fred* starts to dominate and system enters a stable state with *Fred* activated and all the other names back esting level. The system answers the question with the reply: 'Err... Fred.' So, e storage system of table 2.1, this system *does* allow information to be when it has been accessed by content rather than address.

lative activity of the name *Fred* compared to the name *Nick* over the last 50

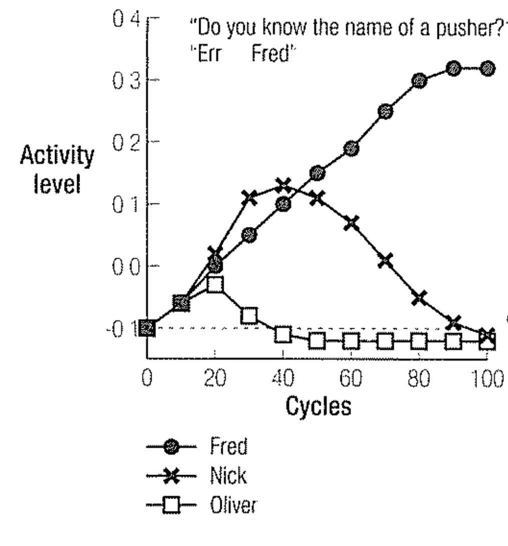


Figure 2.3 To see how the system resp question, the *Pusher* node in the Occu area is held On and activity flows fron through the system The initial activit nodes is set to −0.1. The activity level the nodes in the Name area is plotted function of the number of cycles of ac passing around the system.

cycles demonstrates an important characteristic of models with mutual i between competing responses. When alternatives are equally activated the each other equally and everything is balanced. But once one gets ahead it in others more than they inhibit it. This reduces their activity and thus the which they inhibit the one which is ahead. So it becomes more active and in others yet more. This rapidly results in the one that is a little mo consolidating its position in the lead and completely inhibiting the alternat effect is sometimes referred to as 'the rich get richer' or 'winner takes all'.

Building positive feedback into the system in this way makes it likely system will quickly come to a definite conclusion, even if the difference bet evidence favouring one alternative rather than the other is small. But it m decision process vulnerable to noise. A random disturbance may be magn treated as a signal. This would generally be considered a drawback in a making system but it has one useful consequence. Figure 2.3 suggests that t would always answer the question 'Do you know a pusher?' with the reply so, it would be an indifferent model of human memory. People would give a answer to this question on different occasions, or if asked for an alternativ could provide the names of other pushers. It is straightforward to ach response variability with the model. If random noise is added to the startin levels the system will produce a different answer. Positive feedback ensur node that gets ahead is likely to consolidate its advantage. So a small c starting conditions, or during processing, can make a radical differen outcome. Figure 2.4 shows the result of setting the initial activity level **Person** node to −0.07 rather than −0.1 before activating the *Pusher* node. system answers the question 'Do you know a pusher?' with the reply 'Er

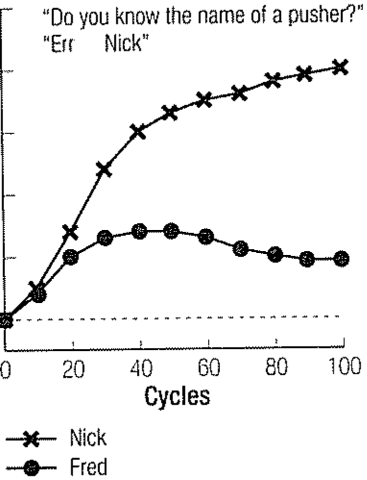


Figure 2.4 The effect of adding noise to the activity levels before asking the question. The initial activity of Nick's **Person** node is set to −0.07 rather than −0.1.

t we know the brain is a noisy system, it is appropriate to model it with an y noisy network. The result is the human characteristic of variability in o the same input.

picality effects in memory retrieval. If the information about the Jets and as stored in the manner of table 2.1 it would not be possible to retrieve the pushers directly, because pusher is not part of the address. But the on could be extracted by sampling addresses at random and giving the ones that turned out to contain the information *Pusher* under **Occupation**. ase any pusher would have an equal chance of being produced as an But this is not how the network behaves when asked to name pushers. As shows, it is more likely to retrieve the names of some pushers than others. etrieve Fred or Nick, but it is unlikely to produce Oliver's name.

n memory retrieval has the same characteristic (see, for example, Rosch you ask people to produce a list of birds (the equivalent of asking the 'Tell me the names of all the pushers you know') most people will include their list but fewer will include Chicken. Some information is more easily for retrieval from memory than other information from the same category. answers which are most likely to be given to this question are the names of ich would be rated as *typical* examples of the category, this result is called, singly, a typicality effect.

ason why the network is likely to retrieve Fred rather than Oliver in reply to est for the name of a pusher can be seen in table 2.2. Pushers are more likely et than a Shark, they tend to be in their 30s, to have been educated to High vel and to be single. So the **Person** nodes which get excited when *Pusher* is send more activity to the *Jet*, *30s*, *HS* and *Single* nodes than to other nodes.

Table 2.2 The pushers

Gang	*Jets* 5	*Sharks* 4		
Age	*20s* 3	*30s* 5	*40s* 1	
Education	*JH* 2	*HS* 4	*College* 3	
Marital status	*Single* 5	*Married* 3	*Divorced* 1	
The prototypical pusher:	*Jet*	*30s*	*HS*	*Single*
Fred	Jet	20s	HS	Single
Nick	Shark	30s	HS	Single
Oliver	Shark	30s	Col	Married

The mutual inhibition between alternative instance nodes with an area m *Single*, *HS* and *Jet* become more activated, and *Married*, *College* and *Shar* less activated. This in turn means that the **Person** nodes connected to *Singl* *Jet* get supported, and the **Name** nodes connected to these become activate **Person** nodes and hence the **Name** nodes connected to *Married*, *College* a do not. Fred is a single, High School educated, Jet; Oliver is a married educated, Shark. So, as a result of Fred's similarity to a *prototypical* pu **Person** and **Name** nodes become more and more active as processing c Oliver's dissimilarity to the prototypical pusher means that his become les active. The result is that when the system is asked to think of a pusher, Fred' retrieved but Oliver's is not. If asked to generate instances from a distributed connectionist nets automatically generate typical instances, people.

Constraint satisfaction in connectionist networks

When activity flows through a connectionist network in response to an in unit influences the state of all the units to which it is connected. If the co weight is positive the sending unit tries to put the receiving unit into the sam activity as itself; if the connection weight is negative it tries to put it into the state. Since all activity changes are determined by these influences, each inp seen as setting constraints on the final state (i.e. set of unit activities) v system can settle into. When the system runs, the activities of individual change in a way which increases the number of these constraints which are Thus connectionist networks are said to work by *constraint satisfaction*.

al final state would be a set of activities for the individual units where all aints were satisfied. The network would then be stable because no unit trying to change the state of any of the units to which it was connected. solution is unlikely to exist because most units are connected to some units trying to increase its activity and others which are trying to reduce it. o way of satisfying both. But if the system can find a state in which any the activity levels of the units reduces the overall number of satisfied s, it will stop changing activities. That is, it will have found a stable state. networks there will be many possible stable states with a different pattern y, each one of which will be reached from a different pattern of input The realisation that these stable states could be viewed as the network's —that is, the set of possible states that it could reach in response to inputs—was an important step in the history of connectionism which will sed further in chapter 15.

nventional connectionist network where knowledge is distributed across its, it is difficult to follow constraint satisfaction at work because it is o see what role any particular unit is playing. In Jets and Sharks it is easy, he concept coding is localist rather than distributed. Each node stands for fiable concept. The constraints on the system are the various facts in table one of which is represented by one of the links in the network. The fact that a Jet in his 40s means that if either of the nodes representing these concepts ed it will try to activate the other, and inhibit the *Shark*, *30s* and *20s* nodes. that there are also both Jets and Sharks in their 30s and 20s means that a t of other, mutually contradictory constraints are influencing the way that m changes the activity level of the units in response to any particular pattern The key point is that the changes in activity on each cycle will increase the of constraints which are satisfied.

tem which works by constraint satisfaction has a number of desirable ristics for modelling human cognition. The main one is that it allows a to be reached by a consensus of evidence, a reasonable fit between input and , rather than requiring an exact match. We have already seen this as a virtue odel of human cognition because the nature of the nervous system requires a f fault tolerance in the information processing system (remember figure 2.1). desirable because of the nature of the input which the cognitive system has with in the real world. Consider what happens when you listen to one ar speaker in a crowded room. The signals arriving at your ear contain the made by the person you are listening to, but superimposed on these are a of sounds from different speakers, the whole thing obliterated from time to bursts of laughter and other noises. And yet, most of the time, what you e is words. The signal you receive bears *some* relationship to a prototypical

representation of the word that you perceive but will be far from an exact m fact that you *perceive* words shows that the word recognition system must b for a *best fit* to the word patterns it has stored rather than for an exact ma effect has been studied in the laboratory with an experimental paradig 'phoneme restoration'. In the original study by Warren (1970) the soun removed from the word '*legislature*' and replaced with a cough. People the to a sentence containing the word '*legi*<cough>*lature*' and were asked w heard. People reported the sentence correctly, adding that there was a cou or after the word 'legislature'. In other words, the perceptual system necessarily give a veridical account of the stimulus, it gives a plausible *inter* of the input, given its knowledge of English words.

The same effect can be seen in the Jets and Sharks system. Figure 2.5 sh happens when the system is probed with a variety of retrieval cues. The fill show what happens when the net is asked: 'Do you know a Shark, in his went to High School, who is single and a burglar?' (i.e. the *Shark*, *20s*, *H* and *Burglar* nodes are all switched On). The circles joined by solid lines activity of Ken's **Name** node and the circles joined by dashed lines the activ next most activated **Name** node. Not surprisingly, the net answers 'Ken'

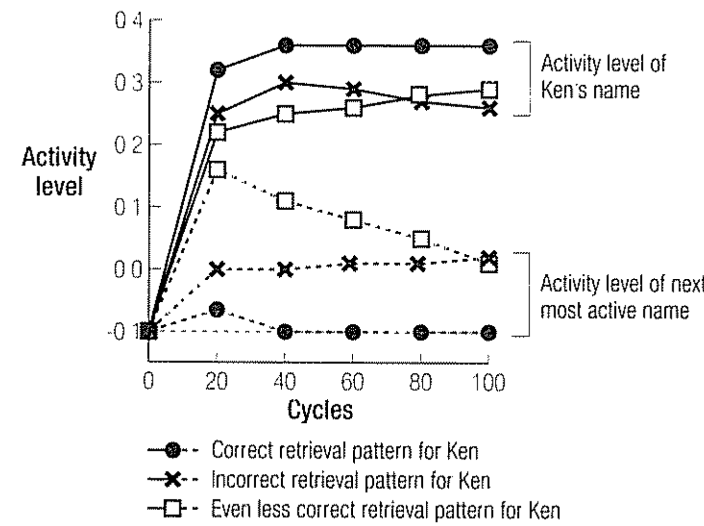


Figure 2.5 Constraint satisfaction in operation. Given a correct description of Ken as a retrie system retrieves Ken's name. Given progressively less accurate retrieval cues, which are, ne closer to a description of him than to anyone else in the database, it still produces his name. In levels:
(●) *Shark* = 1; *20s* = 1; *HS* = 1; *Single* = 1; *Burglar* = 1.
(✖) *Shark* = 1; *20s* = 1; *JH* = 1; *Single* = 1; *Burglar* = 1.
(□) *Shark* = 1; *20s* = 1; *JH* = 1; *Single* = 1; *Burglar* = 0.5; *Pusher* = 0.25; *Bookie* = 0.25

he equivalent of presenting a listener with a clear and unambiguous example ord 'legislature' and asking her what word she heard. The crosses show what when *High School* is changed to *Junior High* in the input activity pattern. out is no longer an accurate description of anyone in the database (the nt of presenting a listener with '*legi*<cough>*lature*'). A system which tried to exact match to the input would fail. There is no person who matches that attern in the database. But the network has no problem. It takes slightly o respond (i.e. Ken's name takes more time to become activated, and does not ch a high level) but *Ken* still comes out as the clearly preferred item retrieved emory. The squares show what happens with an even more ambiguous input. he *Bookie* and *Pusher* nodes have all been activated as well as *Burglar*. As the input pattern is closer to a description of Ken than to any alternative, the makes a clear decision in favour of the response 'Ken'[5].

act that connectionist systems work by constraint satisfaction is the reason ey exhibit fault tolerance. No part of the input uniquely determines the e. The network's response is the best fit it can make between the current input ormation it has acquired in the past. It would be possible to devise a system, n address + contents information storage like table 2.1, which, if given an hat failed to match any stored information, could compute a best fit. er, this would be a time consuming process once possible inputs achieved any of complexity. The parallel distributed computation in the Jets and Sharks se automatically computes a best fit between input and stored information. It continue to do so whatever the degree of complexity of the patterns describing vidual entries without taking any more time.

e is no distinction between 'memory' and 'processing' nnectionist models

eneral point to note about distributed representations is that they blur the tion between memory and processing. Traditional models of cognitive ses often distinguish between 'memory', a store of learnt information, and ssing', operations which enable the system to interpret incoming information. ocessing operations may use information from memory, but the conceptual tion is clear. Indeed, in many models this is made explicit with separate parts model labelled 'memory' and 'processor'. Such models exploit the analogy to conventional digital computers where there are independent systems fo information and for processing it.

There is no such distinction in connectionist models. All the information network has—its memory—is stored in the weights of the connections betw All the processing that the net can do is determined by the same set of weigh

Problems for distributed representations

We have emphasised the advantage of distributed representations ove representations in allowing the system some degree of resistance to dar tolerance of noisy inputs. Given the unreliable nature of the matter which uses for computation, it seems inevitable that it would use distributed re tions. However, there are two properties of human memory which would to be easy to account for with connectionist models but which would be with localist information storage: First, the addition of new information necessarily cause the loss of old. Second, learning can be immediate.

In a distributed system, any new information has to be added to the co which already carry the system's current store of knowledge. To add information the strength of connections must be changed. If this is done i trial, addition of new information is likely to lead to some loss of old inforn a localist system, in contrast, the addition of new information is no prob simply added to new storage locations and does not affect old informatio point, however, we will just suggest that at an intuitive level there are diffe of human learning.

At one extreme it seems clear that some sorts of knowledge can be immediately, without interfering with other information. All young chess p shown the smothered mate sequence with a queen sacrificed to a rook on g1 by mate of the king on h1 by a knight moving from h3 to f2. If you underst you only have to see this sequence once to remember it for ever, despite the you may never have an opportunity to use it in a game. Similarily, if y anything about the British ex-Prime Minister Mrs Thatcher, and were tol nickname at school was 'Bossy Roberts', you would be unlikely to forget it piece of information which you find interesting or amusing, in a domain w already have sufficient knowledge to understand its significance, is lik remembered after a single presentation. And it can be retrieved as a specif information in future, independent of any other facts in the database. It is d believe that such acquisition is accompanied by the loss of any other info Quick, cost-free, addition of new information to existing databases char certain sorts of human knowledge acquisition. It is natural with localist re tion of knowledge—you just add another entry to the database. But it is d

eraction of information in the Jets and Sharks system produces some strange and unpredictable which can only be appreciated by playing with the iac model. For example, figure 2.5 shows that if t long enough the *less* accurate description of Ken produces a stronger preference for his name more accurate description.

it can happen with a distributed system. (That connectionist models *can* one trial learning will be shown in chapter 13 which demonstrates a model ole of the hippocampus in episodic memory formation.)

e other extreme there are many areas of knowledge acquisition, such as g to play tennis or learning to talk, where acquisition of new knowledge is , and accompanied by the modification or loss of previous patterns. As your erve improves or you learn to pronounce the language correctly you *want* to ne aspects of your old response patterns because they were inaccurate. Later it ult to recall when a specific piece of information was added to the database. pattern, where new information is inextricably interwoven with old, occurs ly with a distributed system, but not with a localist one. Distinctions between nt sorts of knowledge representation occur in many models of the cognitive . Most of the connectionist learning algorithms we will look at are more riate for modelling the latter sort of acquisition and representation than the

3 *Pattern association*

This chapter will introduce the architecture and properties of one specifi network, a pattern associator, and the operation of one particular learning Hebb rule. During training a pattern associator is presented with pairs of pa learning is successful then the network will subsequently recall one of the pa output when the other is presented at input. After training, a pattern associato respond to novel inputs, generalising from its experience with similar pattern associators are tolerant of noisy input and resistant to internal damage. capable of extracting the central tendency or prototype from a set of similar e

The first two chapters introduced general principles which are shared by al tionist networks. In this and the next four chapters we will look in detail at ture and properties of a variety of specific network architectures: pattern ass autoassociators, competitive nets and recurrent nets. Networks can be tra variety of ways. In this chapter we will look at one particular learning rule, rule. In chapter 4 we will look at the Delta rule and in chapter 5 at backprop

The examples and the networks in these early chapters have deliberately very simple so that the principles involved in their operation can be seen at may seem that the problems they solve are so trivial that they have little to human cognition. Don't worry. In chapters 8–14 we will look at networ have been scaled up to the point where they can simulate realistic aspects c behaviour. For example, in chapter 8 we will look at a model with roug processing units and 200 000 connections which learns to pronounce monosyllabic words in the English language. In chapter 13 we will look at a episodic memory formation in the hippocampus with about 4000 process and about half a million connections. Exactly the same principles will be at these networks as in the examples which follow.

The architecture and operation of a pattern associato

A fundamental task for the nervous system is to discover the structure of t by finding what is correlated with what. That is, to learn to associate one