CONNECTIONIST MODELLING IN COGNITIVE SCIENCE: 
AN EXPOSITION AND APPRAISAL

by

HENDRIK CHRISTIAAN JANEKE

submitted in accordance with the requirements for the degree of

DOCTOR OF 
LITERATURE AND PHILOSOPHY

in the subject

PSYCHOLOGY

at the

UNIVERSITY OF SOUTH AFRICA

PROMOTER: PROF C PLUG

FEBRUARY 2002
ACKNOWLEDGEMENTS

My sincere gratitude is due to:

Prof. C. Plug for his expert guidance, patience and encouragement throughout the period during which the thesis was written.

Colleagues and friends for useful discussions about many of the theoretical issues surrounding connectionist models and their cognitive implications.

Transnet, and Unisa for a scholarship which allowed me to attend the connectionist modelling seminar which took place during the First Cognitive Science Summer School at Buffalo University.

Marielienne and my children for their support during the preparation of this study.
I declare that Connectionist modelling in cognitive science: An exposition and appraisal is my own work and that all the sources I have used or quoted have been indicated by means of complete references.

H.C. Jones
ABSTRACT

This thesis explores the use of artificial neural networks for modelling cognitive processes. It presents an exposition of the neural network paradigm, and evaluates its viability in relation to the classical, symbolic approach in cognitive science. Classical researchers have approached the description of cognition by concentrating mainly on an abstract, algorithmic level of description in which the information processing properties of cognitive processes are emphasised. The approach is founded on seminal ideas about computation, and about algorithmic description emanating, amongst others, from the work of Alan Turing in mathematical logic. In contrast to the classical conception of cognition, neural network approaches are based on a form of neurocomputation in which the parallel distributed processing mechanisms of the brain are highlighted. Although neural networks are generally accepted to be more neurally plausible than their classical counterparts, some classical researchers have argued that these networks are best viewed as implementation models, and that they are therefore not of much relevance to cognitive researchers because information processing models of cognition can be developed independently of considerations about implementation in physical systems.

In the thesis I argue that the descriptions of cognitive phenomena deriving from neural network modelling cannot simply be reduced to classical, symbolic theories. The distributed representational mechanisms underlying some neural network models have interesting properties such as similarity-based representation, content-based retrieval, and coarse coding which do not have straightforward equivalents in classical systems. Moreover, by placing emphasis on how cognitive processes are carried out by brain-like mechanisms, neural network research has not only yielded a new metaphor for conceptualising cognition, but also a new methodology for studying cognitive phenomena. Neural network simulations can be lesioned to study the effect of such damage on the behaviour of the system, and these systems can be used to study the adaptive mechanisms underlying learning processes. For these reasons, neural network modelling is best viewed as a significant theoretical orientation in the cognitive sciences, instead of just an implementational endeavour.

Key terms

artificial neural networks, attractor memory models, classical cognitive approach, coarse coding, cognition and computation, cognitive science, connectionist modelling, distributed representations, functionalism, mental representation, physical symbol system, Turing machine.
6. **Connectionist models and cognitive theory: An appraisal**

6.1 Posing the problem

6.2 Connectionist models as embodiments of cognitive theories

6.2.1 Using ANNs to test cognitive theories

6.3 A deflationary view of connectionism

6.4 The relevance of connectionist models to cognitive science

6.4.1 Using a subsymbolic approach to break out of the Chinese room

6.4.2 Moving beyond representationalism

6.4.3 ANNs as dynamical systems

6.4.4 Reconciling the connectionist and symbolic accounts of cognition

6.5 A final reckoning

6.5.2 The future of connectionism: Where is it heading?

6.5.3 Concluding comments

**Bibliography**

**Appendix A:** Derivation of the delta rule used in the Widrow and Hoff network

**Appendix B:** Derivation of the backpropagation rule

**Appendix C:** Lakoff's argument against objectivist metaphysics
NEURAL NETWORKS
AND COGNITIVE SCIENCE

During the past two decades, psychologists and other cognitive scientists have increasingly turned to massively parallel computational systems as a methodology for exploring, simulating, and constructing theories of cognitive processes. These models, called artificial neural networks, are not really new because researchers have been experimenting with them since the early 1940s, but it is only relatively recently (in the period 1985-1986) that they have entered mainstream science, largely due to the discovery of powerful algorithms for training complex networks containing several layers of processing units. This breakthrough in designing complex networks had an almost immediate impact on a broad array of disciplines, and ever since these models have been used in many different fields of science, including psychology and cognitive science. Following these developments, there has been a veritable explosion of interest among cognitive researchers who now consider these models to be a welcome adjunct to their repertoire of technological tools (Bechtel & Abrahamsen, 1991).

Part of the allure that the new generation of neural network models hold for cognitive scientists stems from a fairly general consensus that they capture at least some general principles of brain functioning (see O’Reilly, 1996; Anderson, 1995). Some researchers have even suggested that the development of artificial neural networks coincides with the creation of an emerging field called Computational Cognitive Neuroscience (McClelland, 2000, p. xix), which has as its main theoretical aim to understand the brain by building scaled down models of aspects of cognitive and neural functioning. However, despite the widespread use of these models in research applications, their ‘proper’ place in psychology and the other cognitive disciplines is controversial. For example, there are differences of opinion about whether an artificial neural network (henceforth ANN) can be regarded as instantiating a theory of cognition, or whether it should rather be viewed as an implementation of a theory in a particular (parallel) processing system. One reason why there is some confusion about the place of ANNs in cognitive research, derives from the fact that the design decisions governing the development of these models are often dictated by engineering needs - generating the appropriate output - rather than by cognitive or neurophysiological considerations. As a result, the true nature of ANN as cognitive models, and their potential contribution to cognitive science remain unclear.

In this thesis I take an in-depth look at the use of ANNs for modelling cognition, and provide an exposition of these systems. However, my main aim is not expository, but to present a critical examination of their theoretical relevance in cognitive research. Although ANNs have been widely used in psychology and the
other cognitive sciences they have - as already intimated in the previous paragraph - also become associated with a number of controversies and disputes of a somewhat technical nature. I will consider some of these issues in detail in the chapters that follow. In the course of this discussion some of the useful contributions that neural network modelling may make to cognitive research will be highlighted and, just as importantly, some of their shortcomings as tools for theoretical work in cognitive science and psychology will also emerge. As will become evident during the course of the discussion, ANN modelling is still at a relatively primitive, somewhat 'crude' stage of development, but even so they have had a significant influence on both theoretical and methodological work in psychology and the other cognitive sciences. As Hanson and Olson (1991, p. 332) put it: "The neural network revolution has happened. We are living in the aftermath".

This first chapter sets the stage for the ensuing discussion by describing the theoretical rationale for the study, and by presenting a general overview of the themes that will be addressed in the thesis. In subsequent chapters these themes are developed in much more detail. In fact, the main aim of this chapter is simply to sketch a 'birds eye view' of the main themes that will be covered in the thesis, and to tease out some of the main issues attendant on these themes.

1.1 NEURAL NETWORK MODELLING AS A RESEARCH TECHNIQUE

ANN modelling techniques are widely used in scientific and technological applications in academic or research institutions, and also in industry. By way of illustration, ANNs have been used to detect protein structure in chemistry and drug design (Zupan & Gasteiger, 1999), to simulate magnetic spin systems in statistical mechanics (Mézard, Parisi, & Virasoro, 1987), to model nonlinear dynamical systems (Hirsch, 1996), to study various aspects of pattern recognition and classification (Bishop, 1995), to build speech recognition, and orthographic-to-speech conversion systems (Sejnowski & Rosenberg, 1986), to perform fingerprint analyses (Baldi, Chauvin & Hornik, 1995), and to experiment with adaptive computational systems for optical recognition in engineering and artificial intelligence (Poggio & Shelton, 1999).

They are also extensively used in time series prediction, financial modelling, and data mining (Beltratti, Margarita & Terna, 1996; Madridakis, Wheelwright & Hyndman, 1998; Masters, 1995). Visa International makes use of ANN technology to detect fraudulent use of credit cards in various Canadian and American banks, and also employs ANNs to verify the legitimacy of transactions at automatic pumping stations. If a card is stolen there is a window of a few hours before it will be reported and unmanned petrol stations provide a way for screening cards before they are used at shops. An ANN-based risk identification system has been used to spot such transactions and to diagnose their validity (comparing current usage in an information database in which patterns of typical non-fraudulent uses are recorded). The system is reported to have saved Visa an estimated 40 million dollars over a six month period (Holder, 1995). Dutta and Shakhar (1988) used an ANN to predict the classification of corporate bonds with 10 input variables.
describing various aspects of the financial status of the company. The network achieved a prediction accuracy of about 82%, much better than previous statistical techniques which have averaged around 65%.

In view of the proven usefulness of these models in science and industry, there has been a concerted effort to explore their computational and mathematical properties. In some of the most prestigious contemporary journals in this field (e.g. *Neural Networks, Neurocomputation, Connection Science*), the majority of the published articles relate to an investigation of technical aspects of ANNs such as efficient algorithms for training them, theoretical analyses of different learning procedures, and investigations into the mathematical and computational properties of these nets. It is on this level that most of the progress in ANN research has been achieved, and it is here also that the most intensive activity of the ANN research community takes place. Explorations of the specifically cognitive properties of ANNs (i.e. their relevance to cognitive science) have occurred in parallel, but on this cognitive and psychological dimension, theoretical advances have been somewhat slower. This is partly attributable to a number of theoretical and philosophical issues which have arisen around the use of ANNs for cognitive modelling, but also because of the inherent complexity of the cognitive and brain sciences. It is simply difficult to make any theoretical progress in the area, and only a limited understanding has been attained of how cognition is realised in the neural system. Nevertheless, even if researchers are far from ‘cracking’ the proverbial cognitive code, there is a fairly general consensus among psychologists that ANN modelling is an important theoretical direction in cognitive research (e.g. Schneider, 1987).

### 1.1.1 A closer look at a neural network

Perhaps more than any other research fields, neural network modelling is associated with complex terminology. Partly due to the interdisciplinary nature of ANN modelling which has been pursued in a number of different scientific contexts, different terms and ideas have been used to describe these systems, ranging from *neural networks*, *parallel distributed processing systems*, *neural nets*, and *neurocomputational systems* to *connectionist nets*. In many industrial applications, ANNs are simply exploited as mechanisms for statistical computation, but in the behavioural and cognitive disciplines they are used both as a methodology for data analysis, and as a technology for modelling and trying to understand human cognition. The latter is perhaps the main rationale underlying their use in psychology and artificial intelligence. In this ‘cognitive science domain’, ANNs are regarded by some researchers as instantiating a theory of how the mind works, and particularly of how the mind may relate to the brain (e.g. Bechtel & Abrahamsen, 1991). It is useful to distinguish between this general theoretical interpretation of ANNs as a framework for developing theories of cognition, and the development of a specific ANN model (which may or may not be assigned a cognitive interpretation) for a particular application. I will use the terms *connectionism* and *ANN modelling* to refer to the general research enterprise in which ANNs are used to explore, and develop, models of cognition. In contrast, the terms *artificial neural network*
(ANN), connectionist system or just neural net will denote the specific technology (i.e. network models) employed within the broader connectionist framework.

ANNs are computational models based on the notion of parallel distributed processing. In many respects, the resulting computational systems seem very different from a conventional, digital computer. They are firstly constructed from networks of interconnected processing units (called units, or neurons), each fitted with a small amount of local memory. The units function somewhat like neurons (hence the name ‘neural network’) in that they have excitatory and inhibitory links (their ‘synapses’) with other units, and serve to either propagate or inhibit the transmission of ‘neural’ activity in the network. ANNs normally have great potential for parallelism, and transmission of such activity can occur in parallel. This parallel processing capability is commonly taken to be a defining property of ANNs, as the following definition from the DARPA Neural Network Study conducted in 1988 suggests:

“... a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.” (DARPA, 1988, p.69)

In the prototypical case an ANN contains many processing units which are configured in linear arrays (known as layers). There are usually two or more layers of units; an input layer and an output layer, often with one or more hidden layers interspersed between them. The units in the input layer function like sensory transducers, they accept information from the environment. The output units convert the information produced by the input units into an action. Their function resembles the translation of perceptual input into a motor action, as when the sight of a fly triggers a swatting action in most humans. The middle layer is called ‘hidden’ because units in this layer have no direct contact with the world outside the network, their only contact is indirect via the input or output layers. Figure 1.1 on the next page illustrates a multilayer, feedforward network. The network is called a ‘multilayer’ network because apart from the input layer it contains three additional layers; two hidden layers and the output layer.

The network displayed in the figure has a very simple architecture in that it consists of only a small number of units, each of which is only connected to the units in the next layer. The different layers are connected in a strictly feedforward manner with information flowing through the network in one direction only (as indicated by the arrows), from the input to the output layer. More complicated architectures can be designed by increasing the number of units, and by allowing connections to loop back from units in either the hidden or the output layer to the input layer. Neural networks with feedback connections are known as recurrent networks. The point to note is that there are much more complex network architectures than the example shown in the figure. Connectionist nets are rather complicated systems, governed by a large number of parameters, and the variety of different network topologies cannot easily be captured by a single
Each unit in a neural network is assigned an activation level (either a discrete or a continuous scalar value), and a set of weighted connections reflecting its strength of association with other units in the network. These strengths of connections are called the weights, connection weights, or weight coefficients of a network. The input to the network is a pattern of activity over the input units, a vector, and information is processed by passing activation through the interconnections in the network. Essentially activation is passed as a function of a unit's own level of activation and connection weights. Units with positive or excitatory connections tend to raise the activation of the destination units, whereas units with inhibitory connections tend to diminish the activation of units connected to them. These simple neuron-like computations are performed in parallel.

In some of the discussions later in this thesis, connectionism will be contrasted with other approaches in terms of its assumptions about cognitive architecture. The concept of an 'architecture', as it is used in cognitive research, is somewhat difficult to grasp. It originated in computer science (e.g. Tanenbaum, 1984), where it denotes the arrangement of computational components such as a central processing unit, memory components, and display utilities, that are associated with a specific computational device, and define its information processing properties. The concept is not to be confused with a description of the information processing procedures employed in the execution of a particular task. Instead, it denotes a more abstract level of description focussing on the underlying computational system, or virtual machine.
(i.e. a machine modelled by software), that runs the program, and describing a cognitive architecture amounts to a functional specification of the relevant cognitive, virtual machine. A cognitive architecture denotes all the structures and mechanisms of the cognitive system that are genetically specified, that are not modifiable by experience, and that determine the gross division of the cognitive system into processing modules, communication pathways, and storage mechanisms (Pylyshyn, 1984). As will become clear in the discussion that follows, ANN models are assumed by some researchers to instantiate a different cognitive architecture from the rule-based, serial processing mechanisms postulated under the mainstream, 'classical' approach in cognitive research (see e.g. Bechtel & Abrahamsen, 1991).

A concept that will frequently appear throughout the rest of this text is representation. An intelligent system, whether natural or artificial, can only learn, remember or recognise if it is capable of storing an internal representation of what it has learned. Artificial intelligence researchers and cognitive psychologists have made use of a variety of different tools and techniques to represent information such as semantic networks, production systems, logic, scripts, and frames (Markman, 1999). ANN models are based on a different approach in terms of which the weights, that is, the strength of connections between units in the network, encode whatever knowledge it has acquired.

Information is represented in an ANN in the weights and activation patterns in the network. The information may be represented in a localist manner in which each unit has an individual semantic (i.e. symbolic) content, or in a distributed manner. In the latter case, the information associated with a concept or word is spread over a number of units, so that a single unit may participate in the representation of different concepts. The concept is therefore encoded in a distributed fashion because a number of different units contribute to its semantic representation. The idea of distributed representation of information is not very intuitive, and will be discussed in more detail in Chapter 4. It suffices to grasp at this stage that 'distributed' is used to convey the idea that many different units may participate in the representation of a particular piece of information. To highlight the use of such distributed storage of information in a particular ANN, I will refer to such examples as PDP systems or PDP nets (which stands for "Parallel Distributed Processing nets").

ANN models are usually developed using a simulator, which is a software environment that enables a researcher to design and explore an ANN model on a conventional computer. Various commercial and freeware ANN simulators are available, some of the best ones are listed by Sarle (2001). ANNs are also sometimes run on special purpose hardware called accelerator boards, and the development of specific ANN chipsets are underway. However, these hardware implementations are still quite expensive, and by far the most common way of experimenting with ANNs is to use a software simulator, for constructing, and training these systems (see also Murre, 1995 for a discussion).
1.1.2 Learning and generalisation in neural networks

One of the main reasons for the current interest in ANNs stems from the fact that they learn from experience. Essentially, a neural network learns by exploiting procedures that gradually adjust weights and activation values in the direction of more appropriate output. Different learning algorithms exist, but typically learning is construed as a form of gradient descent in a multidimensional weight space (the pattern of weights characterising the connectivity among the units in the network). For example, the generalised delta rule, or ‘backpropagation rule’ discovered by Rumelhart, Hinton, and Williams (1986) is a popular learning method for training multilayer networks. During training an input training set is presented to the net over many (typically several thousand) training iterations or cycles during which the errors in the network’s performance are adjusted towards more appropriate output. At the end of training the network’s performance is evaluated. If the data on which it was trained exhibit some kind of pattern, the network will typically be capable of generalisation. In other words, given new input taken from the domain on which a network was trained, the net will in some cases be able to compute output values for these data. If this happens the network is said to have learned.

There are many issues and controversies relating to the extent to which an ANN is capable of generalisation, and the significance of their learning capabilities for cognitive science (some of these issues are addressed in Chapter 6). Still, once an ANN has discovered a set of weights that allow it to generalise, the model will in many cases be able to repeat a pattern of activity at some later time, or produce a pattern of activity when given some portion of it as a clue. In a sense these systems program themselves, and they can learn to do quite complicated tasks in this way. For example, in Chapter 2 it will be explained that ANNs endowed with a hidden layer are in principle (there are some practical problems) capable of learning to approximate any computable input-output function (Hornik, Stinchcombe & White, 1989).

As mentioned above, ANNs learn by being trained on representative examples taken from the domain concerned. The networks are usually trained with a ‘teacher’ who gives the network feedback on the output that it computes. This type of learning is called supervised learning. In some cases the teacher does not indicate the correct target values of the mapping to learn but only informs the network whether it is on the right track. This type of learning is called reinforcement learning and has been successfully applied in game theory (see Sutton & Barto, 1998, pp. 261-268). ANNs can also be trained without a teacher acting as an intermediary. This is called unsupervised learning, and typically involves a situation in which a network is simply exposed to relevant data and has to discover any patterns existing in the data on its own. There are some questions though about the effectiveness of applying unsupervised training to higher level cognitive processes, such as language, because Gold has proved that it is impossible to induce the grammar of human language without feedback (Gold, 1967).
Some of the learning algorithms used in ANN modelling are similar to existing statistical techniques. For example, backpropagation is conceptually similar to a form of nonlinear, logistic regression (Rumelhart, Durbin, Golden & Chauvin, 1995). The primary difference between ANN learning techniques and other statistical procedures relates to the mode of data processing. In most statistical techniques, the data is presented to the system one time only during which the analysis is carried out. In the neural network paradigm, each individual item in the data set is presented to the net iteratively until the network manages to identify the best-fit pattern of input-output associations. Moreover, the whole process takes place in parallel, because communication between the different layers of units (or ‘neurons’) in the network is instantaneous. It is this parallel processing capability which endows the network with a certain brain-like character, so that it can be said to emulate the massively parallel processing features of the brain. In addition to its brain-like properties, the parallel and distributed processing mechanism underlying connectionist nets offers several advantages which are discussed under the heading “the connectionist alternative” further down.

To summarise: the distinguishing feature of neural network models is that they acquire their weighted connections through exposure to data from a domain, and that they make use of a parallel processing architecture. Knowledge in these nets is implicitly represented by the coefficient weights in the neural connections linking the layers of neurons. An interesting aspect of such systems, from a psychological perspective, is that they exhibit self-organising properties. They are able to uncover statistical patterns inherent in some environmental data, and can adapt to these patterns under certain conditions. The learning and adaptive mechanism underlying these networks is probably one of the main reasons why psychologists and cognitive scientists involved in modelling human learning abilities have become interested in them.

1.1.3 **ANNs have many degrees of freedom**

Although neural networks exhibit some learning properties, one should not exaggerate their similarities to real organisms. These networks are artificial systems, their neural properties are vastly simplified models of some characteristics of living organisms and they have only been tested on relatively small-scale, toy problems. In fact it is probably safe to say that the neural network enterprise at its current stage of development offers only crude, vastly simplified artificial replicas of real brains. They have not yet attained the power of the brain of an insect, and are far removed from presenting realistic models of human brains. At best, the neural network approach offers a general framework, a toolbox or a set of techniques for modelling some aspects of psychological processes, and perhaps a formal language for generating psychological theories (although even this latter aspect is questioned by some cognitive scientists). One problem with evaluating the enterprise is that it is difficult to pin these nets down, setting up a neural network model involves many degrees of freedom, and very different network topologies can be designed for the same training set (the patterns to be learned). The design of a neural network is governed by many
parameters such as network configuration, activation function, and learning rate. These parameters are largely determined by the properties of the statistical patterns inherent in the relevant content area, and cannot easily be set by a researcher on an a priori basis.

Because there are several design choices associated with the development of a network model for a particular application, one should be cautious about assigning a theoretical interpretation to any particular connectionist network. In fact, the theoretical relevance of specific ANN models raises rather problematic issues such as whether these models should be viewed as just ‘black boxes’ or whether they yield a set of principles and ideas that can lead to novel methodologies for cognitive science. In general, it is probably safe to say that ANN modelling is more like a research paradigm, it constitutes a framework for theorising but it is less clear whether a given network constitutes a theory. Note further that because there are so many degrees of freedom associated with the development of neural networks, an investigation into the potential of the connectionist framework for generating psychological theories must necessarily be pitched at a rather abstract level.

1.1.4 Example of an ANN model: Hinton’s family tree simulation

To make the description of ANN modelling a little more concrete, it is useful to consider a simulation developed by Hinton (1990) in which he tested the ability of ANNs to discover hierarchical, part-whole relationships in a tree-structure. He constructed an ANN to learn the relationship among people in two isomorphic fictional family trees, one English and one Italian. The information in these trees can be represented as triplets of the form (Person 1, Predicate, Person 2) where the predicate indicates the relationship between the two persons. Thus, the triplet (Christine, Wife, Andrew) captures the relationship between two of the ancestors at the top of the English family tree at the top of Figure 1.2 (on the next page).

Twelve predicates were used to describe the family relationships, namely father, son, brother, husband, uncle, nephew, and their female equivalents. During the learning phase, the net was presented with the first person and the relationship as input, and the appropriate second person completing the triplet was used as output. Any discrepancy between the target or desired output, and the actual output generated by the net was treated as error, and used to adjust the weights in the network.
Figure 1.2 A portion of two isomorphic family trees used in Hinton’s ANN simulation

The network used by Hinton is shown in Figure 1.3 below. Each person in the tree is represented by the activity of a single unit at input or output, and each of the 12 relations are likewise represented by a single unit in the input layer. In addition there are several layers of hidden units to enable the network to construct its own internal representation of the individuals in the two families, and their relationships to one another. There are 6 hidden units to form an internal representation of person 1, and another 6 to encode the nature of the relationship. These units feed into a layer of 12 hidden units which construct a representation of the combination [{person 1}, {relationship}].

The network was trained on 100 of a total number of 104 relationships in the two families. After training it was tested by activating the inputs representing one person and the predicate and then establishing whether the network could correctly produce the target output.
Hinton found that the net could correctly answer questions about the 100 relationships it was trained on, but it also succeeded in deducing the four relationships that had not been shown to it during the training phase. It is easy to underestimate the significance of the network’s accomplishment, so this merits commentary. Suppose that the fact that Alfonso is Lucia’s son is one of the facts that the network had not received during training. When it is presented with input information corresponding to the question “Who is Lucia’s son?” activity would pass down every connection leading from the inputs Lucia and Son. If the network was a single layer, positive connections would have developed during training from the Lucia input unit to all the output units corresponding to people she was related to. These would include Marco and Emilio because the network was taught that Marco is Lucia’s husband and that Roberto’s son is Emilio. There will also be a positive connection from Marco to Alfonso because the net was informed that he is Marco’s son. However, the network would not get the right answer to the question “Who is Lucia’s son?”, if the output was driven by the strength of associations between input and output units alone. Both Marco and Emilio would get more excitation than Alfonso if the units Lucia and son were activated. The question can only be answered correctly by making a deduction, based on an awareness of transitive family relationships, of the following kind: Marco is the husband of Lucia, and his son is Alfonso, so Alfonso must be the son of Lucia as well. The fact that the net was capable of answering questions correctly about family relationships which it had not been taught, suggests that it was not just memorising relationships by rote, but actually developed some knowledge of the hierarchical structure of the family tree. Evidently, it needed to discover enough about the structure of the tree to be able to draw appropriate inferences about
family relationships in order to answer the questions correctly.

An important point to be gleaned from a consideration of this example is that constructing an ANN simulation of a cognitive process is not a simple automatic process of simply exposing the net to a content domain. Instead a researcher has to design a network for a particular application. In constructing the family tree simulation, Hinton had to find a way of representing the salient information about family relationships in a format that is accessible to the ANN model, and he had to develop a network configuration for the learning task. More generally, designing such a simulation entails making decisions pertaining to:

- the encoding of the input information, because the data must be prepared and presented in such a way that it is suitable for training an ANN. This aspect of ANN modelling is called ‘preprocessing’ the input data (see Masters, 1993, p. 253);
- the network structure used for training on the data set;
- the learning procedure and parameters set for training the network, that is, the ‘components’ of a network;
- when training should stop, and how the network will be tested to verify that it has learned; and finally
- the simulator, and hardware used for modelling the process.

These decisions about the ANN design process, determine how well the network will learn, and even whether it will learn at all. Nevertheless, as the example illustrates, an appropriately designed ANN typically will exhibit some learning and generalisation capabilities. The learning situation associated with the network in the example, is in broad outline psychologically realistic. One can imagine a situation in which a young girl learns about a family tree by systematically being told about the relationships between particular individuals in the family. In a first, memory-based phase she just remembers how some members of the family are related, but once she has accumulated enough information she will be able to infer some additional facts. However, even though the example may have some psychological relevance, one should be careful of reading too much into it, and of ‘anthropomorphising’ the network. This is not the time to consider the many theoretical issues surrounding ANN learning, instead, the example is merely intended to show that ANNs are capable of a limited form of learning and generalisation.

1.1.5 The hype associated with ANN modelling

As already noted, the area of ANN research is both a scientific domain and an application area devoted to the development of computational models that implement a form of parallel processing and have learning capabilities. These models have a long history in psychology, but their current popularity in psychology
and cognitive science can largely be attributed to the pioneering work of two cognitive psychologists, Jay McClelland and David Rumelhart, and an interdisciplinary research group called the parallel distributed processing (PDP) research group. Their combined efforts culminated in the publication of a two-volume manifesto (Rumelhart, McClelland & the PDP Research Group, 1986; McClelland, Rumelhart & the PDP Research Group, 1986) in which the foundations of the neural net approach to cognition were laid out. The PDP books brought neural network techniques to the awareness of the larger scientific community and led to a general interest in the use of ANNs for a variety of different modelling tasks.

The impact of these books on the cognitive science community was almost immediate. In fact rumour has it that the first 6000 copies of the two volumes were sold out while they were still in print (Schneider, 1987). Such a demand is quite unusual for books that are reasonably technical in content, and concerned with, at the time of publication, a relatively obscure area of science. In many respects, the PDP books changed the face of neural network research, and their legacy is now manifest in teaching and research programmes in neural computation at various universities and research institutes, and in a broad array of neural network products marketed by private companies for applications ranging from forecasting and data mining, to optical recognition, speech processing, and nonlinear modelling applications (see Heath, 2000). Furthermore, at least 20 international journals and five annual conferences are dedicated to the dissemination of research on neural networks (Sarle, 2001).

Cognitive psychology is one of the primary disciplines in which ANN research is now conducted due to a strong conviction among some psychologists that these models yield a computational framework that is particularly conducive to the modelling of psychological processes. Indeed, Schneider (1987) goes so far as to proclaim that the advent of these models has brought about a "paradigm shift" in psychology. There are at least two reasons contributing to the surge of neural networks in the cognitive sciences. Firstly, many researchers are attracted to neural networks because they hope that these systems can be used to explore the neural basis of cognitive processes, and that this methodology might eventually lead to insights that will help to bridge the gap between neural and cognitive levels of explanation (Dinsmore, 1992). Secondly, some cognitive researchers are dissatisfied with the prevailing symbolic, logic-based approach to the development of computational models of cognition, and regard neural network modelling as an alternative formalism which might avoid some of the pitfalls of logic (see Chater & Oaksford, 1991). The cognitive science framework is discussed in more detail in the next section.

Despite the apparent usefulness of ANNs in the cognitive sciences, there is, unfortunately, also a considerable amount of hype about them in the popular press. As Dewdney (1997, pp. 80-81) points out:

"The image of neural nets as miniature brains was pumped in the papers, science magazines, on public TV, and even in a few Hollywood movies. Neural network proponents (also called
connectionists) did not rush to their phones to disavow these wild claims.

Most responsible ANN researchers say something to the effect that research into artificial neural nets was inspired by biological neural nets, but they do not make wild, over-blown claims about the intelligence of neural nets. Hecht-Nielsen (1990) even has a section called "Hype" disavowing the "wild speculations" that some researchers have perpetrated, and Mark Seidenberg has a disclaimer on his web page in which he distinuates himself from any hyperbolic assertion about the capabilities of ANNs. In addition Warren Sarle who maintains the FAQ (frequently asked questions) for the usenet group comp.ai.neural-net, has compiled a paper in which a simile of parachuting kangaroos is used to describe ANN learning procedures (Sarle, 2001). The simile was initially conceived to explain ANN learning techniques to readers unfamiliar with calculus, but has grown in popularity because it helps to avoid some of the problems inherent in descriptions of these systems as artificial brains. As will become increasingly evident in the later chapters of this thesis, the ANN approach is certainly not a universal panacea for all the issues, problems and the general theoretical messiness that plague the cognitive sciences in general (and psychology in particular). There are many problems and difficulties associated with the use of ANN techniques, and the existing models are extremely primitive in regard to real brains. There are also many technical difficulties in interpreting their status vis-à-vis the standard computational models in cognitive science.

1.2 NEURAL NETWORKS AND COGNITIVE SCIENCE

The brief exposition presented above merely conveys the flavour of connectionist research, a systematic discussion is reserved for subsequent chapters when the principles governing the performance of ANNs will be addressed in greater detail. In this section I will consider the research framework underlying connectionist theorising in cognitive science, as this provides the background against which much of the later discussion will take place. Before discussing the theoretical rationale of the study, it is useful to say something about cognitive science, and how ANN modelling fits into this research domain.

Cognitive science is the interdisciplinary study of mind and intelligence, embracing a variety of disciplines such as philosophy, psychology, artificial intelligence, neuroscience, and linguistics. Its intellectual beginnings date back to the mid-1950s when researchers in several fields began to develop theories of mind based on complex representations and computational procedures. During the 1970s its organisational structures, such as the Cognitive Science Society, were formed and the journal *Cognitive Science* was established (See Gardner, 1985; Eysenck & Keane, 2000). At present, more than fifty universities in North America and Europe have created cognitive science programmes, and many others have instituted courses in cognitive science. The cognitive science cluster of disciplines emerged following the general demise of the behaviourist approach during the 1960s when psychologists began to realise that behaviourist explanations were simply not powerful enough to deal with aspects such as language and reasoning. During
this period researchers became more and more convinced that an adequate description of human cognition would entail describing the internal mechanisms of the mind. The movement away from behaviourism was inspired by developments in the theory of computation (this will be discussed in Chapter 3), was influenced by ideas borrowed from information theory, and drew from empirical results in verbal learning and memory research (Lachman, Lachman & Butterfield, 1979). However, probably the most significant contribution to the development of the cognitive approach was Chomsky's work on language. Chomsky (1965; 1980; 1981) showed that it was possible to present a detailed and systematic exposition of the cognitive principles underlying language, and that this could be achieved with the aid of standard scientific and mathematical techniques. He also presented convincing arguments that language learning and language understanding cannot be satisfactorily explained using the purely associative approach adopted by behaviourist psychologists such as Skinner (see Chomsky, 1959). Instead, he suggested that any account of the generative mechanisms underlying language use (i.e. the ability of speakers to produce and comprehend novel utterances), necessarily entails considering the internal mechanisms of language.

Cognitive science is now an active research programme, which has both a scientific and technological aspect to it. The scientific aspect is revealed by a large collection of empirical data about human cognition (Eysenck & Keane, 2000), and about the neural underpinnings of the cognitive system (Cotterill, 1998). The technological aspect is mainly reflected in attempts to create intelligent machines, such as AI systems, software agents, robotics (a humanoid robot called “COG” is being developed at MIT), and of course, ANN applications (see Nilsson, 1998, pp. 407-412). There is a fairly general consensus among researchers about the broad aims of the cognitive enterprise, and consequently that the twin goals of understanding the mind (and its relation to the brain) and the construction of smart devices define cognitive science as an academic field (see Thagard, 1998). However, it should be stated right from the outset that even if most researchers accept the general goals of the approach, cognitive science is far from a unified paradigm. There are considerable differences in opinion about whether its research goals can really be fully achieved (particularly the one relating to intelligent machines), and in the specific techniques and theoretical (i.e. philosophical) framework that should be used to guide research in the domain. In fact, much the same situation holds in cognitive science as Boden (198, p. 71) observed about psychology:

There is no normal science, in Kuhn’s sense, no universally accepted way of conceptualizing theory and experiment in psychology. It is as though opposing armies were fighting battles in order to win the right to define the nature of war.

In cognitive science much of the dispute concerns around the way in which the mechanisms associated with cognition should be conceptualised and studied. There is a fairly general agreement among researchers in the field that the mind can be viewed as an information processing system (Eysenck & Keane, 2000, p. 4), although some do not even grant that much (Edelman & Tononi, 2000). However, beyond a fairly general
acceptance of the ‘information processing’ as a metaphor for thinking about the human cognitive system, there is not much that bind researchers together. For example, there is disagreement about what exactly such a commitment to information processing means, and how it translates into research practice.

Dennet (1986) presented an interesting, and amusing, sociological caricature which illustrates some of the issues that divide the cognitive community. The caricature stems from a conference at MIT, during which Jerry Fodor is supposed to have characterised the views of his opponent as “West Coast”. This was indeed a “weighty indictment” because like most MIT researchers, Fodor considers Boston to be the centre of the universe (Dennet, 1986, p. 63). When someone observed that the accused researcher was from Pennsylvania, Fodor was undeterred, arguing that just as anyone moving away from the North pole, is going south, just so any movement away from MIT is in a western direction. Spurred on by Fodor’s twisted logic, Dennet drew up a “logical geography” of the cognitive science community. The geography has MIT at its epicentre, and then projects away from MIT in a circular ‘western direction’ to other American universities. In Dennet’s depiction, the geographical area is laid out in terms of adherence to different versions of a computational metaphor. Researchers in MIT practise what he calls “high church computationalism”, endorsing a strict form of computation closely tied to mathematical logic. As one moves away from the centre, the commitment to the computational metaphor weakens and researchers become more fuzzy in the connotations they attach to the idea of ‘information processing, eventually ending with the “zen holists” such as Rumelhart and the PDP group. In the latter approach, information is equated, in a holistic sense, to patterns of activations in large networks of processing elements.

In Dennet’s (1986) characterisation cognitive research is construed in terms of a polarisation of researchers between two interpretations of cognition. Under both interpretations cognition is a form of information processing, but the notion of information processing is interpreted differently. Under the ‘high church’ view it is equated with a computational system operating on symbols. This view is often referred to as the classical theory of cognition, or just the classical approach. The alternative ‘zen holist’ approach is the connectionist view of cognition in which activity patterns among units in a network (instead of symbols) constitute the main elements of the information processing system. The two approaches are discussed in more detail below.

1.2.1 The classical research framework

The classical approach is essentially the mainstream, information processing perspective adopted in cognitive psychology and cognitive science more generally (see Eysenck & Keane, 2000, p.3). It also includes the symbolic, logic-based research approach in artificial intelligence that Haugeland (1985) refers to as “GOFAI” (Good, Old-Fashioned Artificial Intelligence). The idea of information processing is so fundamental to the classical approach, that it takes on the status of a paradigm in the Kuhnian sense.
(Lachman, Lachman, & Butterfield, 1979; Schneider, 1987). The theoretical motivation for this information processing conception of cognition stems in part from seminal work in the theory of computation by inter alia Alan Turing and Alonzo Church (see e.g. Boolos & Jeffrey, 1989) which will be reviewed in Chapter 3. An important concept emanating from work in computational theory is that we can abstract from the physical operation of a complex system to focus on its functioning as a computational device. It is this aspect that motivates psychological applications of the concept of computation. Essentially the assumption is that cognitive explanations can be pursued by abstracting away from the physical, neural system and concentrating instead on an information processing, algorithmic level reflecting the brain's functioning as a cognitive system.

The guiding assumption underlying the classical perspective in cognitive science is that the mind is an instantiation of a computational system, and consequently that cognitive theorising should be directed at clarifying the way in which the mind (i.e. the cognitive system) executes the information processes associated with different psychological tasks. It is assumed that information processing operations take place on mentally represented knowledge structures. The central dogma in this field of research is that the mental representations are symbolic in character, and that cognition involves a process of symbol manipulations analogous to the way in which such operations are carried out by digital computers. The operations are assumed to be rule-like, logical operations involving actions such as moving, concatenating, substituting, or searching for symbols or symbol combinations. The approach is probably best characterised in terms of two basic theoretical postulates:

- Brains as well as digital computers are physical symbol systems. In both cases the computational properties of the system derive from the manipulation of tokens (cognitive representations of symbolic information, and electronic bits of information respectively) which are fundamentally symbolic in character, because they have a referential function (i.e. they denote objects, events or situations). The symbolic capacity underlying these systems is assumed to be fundamental, and classicists would argue that there is little hope of making progress in understanding of these complex computational systems unless their representational capacities are taken into account.

- The syntactic processes defining these computational systems are algorithmic in character, they can be captured in explicit procedures or rules, which specify the actions to be carried out on the symbols. These rules may be very complex in nature, so that classical researchers typically adopt a divide-and-conquer strategy in terms of which they analyse complex cognitive processes such as memory or perception in terms of their component parts. For instance, classical researchers conceptualise a perceptual process such as object recognition in terms of subprocesses involving feature identification, semantic classification, and naming, and accept that these may all run in parallel (Eysenck & Keane, 2000).
To summarise: classical theories of cognition try to elucidate the algorithmic processes associated with psychological functions. The assumptions are that the mind can be conceptualised as a black box which receives information and produces outputs (e.g. overt behaviour) and that the task of the cognitive scientist is to clarify the nature of the computational processes underlying the input-output mapping that this black box performs. Of course the algorithms characterising the information processes of the mind are ultimately implemented in the neural hardware of the brain, but as a rule theorists operating within the classical tradition are not particularly interested in the physical instantiation of mental algorithms in the human brain. Their concern is with the brain as a symbol-processor, and their research is mainly directed at elucidating the cognitive and computational (rather than the neural or physical) aspects of the brain’s functioning.

1.2.2 The connectionist alternative

Cognitive modellers using neural networks draw from an explanatory framework which is rather different from the symbolic, serial processing approach prevailing in the classical paradigm. The explanations emanating from connectionist research are pitched at a lower level, closer to the level of physical systems, and it is proposed that the massively parallel processing units in such a network may provide a new framework for understanding the nature of the mind and its relation to the brain (Rumelhart, 1992). Furthermore, several properties of ANN models suggest that connectionism may offer a useful picture of the nature of cognitive processing. ANNs exhibit robust flexibility in the face of the challenges posed by the real world, because as explained below, noisy input or destruction of units in an ANN system causes graceful degradation of function. Moreover, ANNs are particularly well adapted for problems that require the resolution of many conflicting constraints in parallel. There is ample evidence from research in artificial intelligence that cognitive tasks such as object recognition, planning, and even coordinated motion present problems of this kind. Although classical systems are capable of multiple constraint satisfaction, connectionists argue that ANNs models provide much more natural mechanisms for dealing with such problems. In addition, Researchers such as Chater and Oaksford (1991) have identified a number of "allures" (i.e. particularly attractive properties from a cognitive point of view) of ANNs. Some of these allures are listed below.

☐ There are many issues and controversies relating to the extent to which an ANN is capable of generalisation, and the significance of their learning capabilities for cognitive science (some of these issues are addressed in Chapter 6). Still, once an ANN has discovered a set of weights that allow it to generalise, the model will in many cases be able to repeat a pattern of activity at some later time, or produce a pattern of activity when given some portion of it as a clue. In a sense these systems program themselves, and they can learn to do quite complicated tasks in this way.

☐ ANNs are at least in some general sense neurally plausible. The brain is essentially an organ that
processes information in parallel, carrying out basic processing very rapidly, within the order of a few milliseconds. It is known that each neuron takes several milliseconds to fire. Most cognitive tasks can be decomposed into a sequence of logical steps each consuming processing time. If these steps were performed serially, the rate at which basic processes are performed should be the sum of the processing time consumed by the various subprocesses, and thus slower than the millisecond time constraint. Thus, given the rather slow rate of neural firing, the very fast speed with which cognitive processes are performed can only be accounted for if we assume that the component parts are executed in parallel (Feldman & Ballard, 1982). Connectionism exploits mechanisms for parallel distributed processing and thus furnishes a more plausible framework for theorising about neural processing than the typical serial model flowing from work in the classical tradition.

The operation of a classical model is typically discrete, deterministic: cognitive rules only apply if their preconditions are satisfied, and the elimination of even a single rule can cause the system to crash (i.e. stop functioning). As a result, the computational models emanating from work in the classical tradition tend to be somewhat brittle because they are highly susceptible to damage. In contrast the functioning of an ANN system is much more resilient because its information processing capacity is based on a large number of small processing units acting in concert. If some of these units are destroyed, the system does not crash but exhibits a form of graceful degradation. Its computational abilities gradually diminish in accordance with the total number of units destroyed, reflecting the severity of the lesion, and the system can tolerate a degree of damage before it ceases to function adequately. This property of graceful degradation exhibited by connectionist systems resembles the ability of the brain to endure neural injury while retaining some of its functions and significant processing capacity.

One of the main contributions of connectionist research is to show how the ostensibly rule-governed aspects associated with cognitive processes can emerge from a fundamentally parallel distributed processing system at the micro level. Hence, connectionism paves the way for exploring the representational capacities of parallel-processing systems, like the brain, in which the computational abilities are seated in a large number of small processing units spread out over an interconnected network.

Human memory is content-addressable, one can access a memory trace using some part (its content) of the memory as a retrieval cue. This content-based retrieval is different from other familiar forms of storage such as telephone directories or traditional computer storage mechanisms. In these systems, information retrieval is based on a memory address and the only way to access information is to locate the address where it is stored. In a dictionary a word is stored on the basis of its spelling which places it at a specific location in an alphabetised list of words. ANNs also allow
content based access because of the distributed way in which information is stored in these systems. The content-based retrieval mechanisms associated with ANNs will be considered in detail in Chapter 4.

From the discussion above it appears that there are reasons to suggest that ANN models instantiate a rather different form of processing than that assumed under the classical approach. However, the exact role of ANN models in cognitive research is not clear at all. ANNs are, at least today, difficult to apply successfully to problems that concern manipulation of symbols which are important in areas such as natural language understanding. Moreover, there are differences in the conception of what exactly an ANN is. Most connectionist researchers conceive of ANNs as essentially computational systems employing parallel distributed mechanisms. A rather less common view, but one which is gaining grounds among researchers, is that some ANNs (specifically those endowed with recurrent connections) are not adequately captured using the language of computational systems, but that they are more aptly described in terms of the concepts of dynamical systems theory. In this latter perspective the focus is on the temporal dynamics of a net as it moves through a state space and aspects relating to its learning and processing are explained with recourse to notions such as 'attractors' and 'trajectories' instead of the typical computational vernacular of 'search', 'representation' and 'symbol manipulation'. This lack of a clear consensus about how exactly ANN models fit into the theoretical framework of cognitive science is the main rationale for the present study. A detailed motivation is presented in the next section.

1.3 MOTIVATION FOR THE STUDY

In order to set the stage for the ensuing discussion, it is fitting to demarcate the scope of the study presented in this thesis by isolating the themes that will be developed throughout the rest of the thesis, and also to offer at least some preview of the interpretation of ANN modelling that will be developed in the study. Even if the approach appears to have become a popular technology, or 'toolbox' among cognitive researchers it is far from clear that it is a real competitor of the traditional computational approach, and that connectionist modelling actually advances our understanding of cognitive systems. Many cognitive researchers would argue that connectionist nets are best viewed as a technology for exploring the neural basis of cognition without necessary shedding any more light on the cognitive aspect than conventional symbol processing models of cognition. It is not at all obvious that these models are relevant to psychological theories of human learning and cognition. In fact, many ANN models have their origin in engineering, computer science, and statistics and may have considerable general scientific and technological appeal, but not necessarily any specifically psychological relevance.

Moreover, it is relevant to note that the idea that ANNs can be used for modelling aspects of cognition is neither new, nor particularly interesting on its own. Various other researchers have already argued that
connectionism presents a set of modelling techniques that can be used to enhance our understanding of the functioning of the human learning and memory systems (e.g. Hanson & Burr, 1990). In addition, an existing set of applications of connectionist ideas to psychology strongly suggests that AANs may be used to explore and uncover some of the general principles underlying neural and cognitive processing (see Levine, 1983). However, although connectionist models have become increasingly popular in cognitive science, and although there is a slow but steady increase in our theoretical understanding of the mathematical principles underlying these networks, there are at least three general problems which preclude the widespread adoption of connectionism as a framework for psychological theorising.

- Firstly, most of the connectionist models developed in the mid 1980s were directed at toy problems (i.e. small scale applications). Some researchers have complained that it is not clear at all whether these models can be scaled up to deal with larger, psychologically realistic aspects of cognition (Minsky & Papert, 1988).

- Secondly, some eminent cognitive theorists have argued persuasively that ANNs are best regarded as implementations of classical (symbolic) models (e.g. Broadbent, 1985; Fodor & Pylyshyn, 1988). The implication is that connectionist networks are simply pitched at a lower level than extant symbolic models of cognitive functions, but do not necessarily bring anything new to our understanding of these functions. The point of this criticism will become clearer when we consider the Church-Turing thesis in Chapter 3.

- Finally, many cognitive scientists reject connectionism as simply associationism in a new guise. It is generally acknowledged in the cognitive science community that associationism has serious drawbacks as a theoretical approach to higher-level cognition, particularly reasoning and language processing. Hence by equating connectionism with associationism, the implication is that it has severe limitations as a model of those aspects of cognition where associationism has failed (e.g. language processing). Indeed one of the requirements of any theory of cognition is that it should account for the creativity underlying higher level cognitive functions, and associative approaches have difficulty in dealing with the creative aspect of cognition.

It would therefore appear that there is not much consensus even among connectionist researchers as to what exactly connectionist nets are, how these systems should be conceptualised, what concepts and terminology are appropriate to describe their behaviours and particularly how they relate to the conventional information processing models developed in the heyday of the computational approach.

The main aim of this study is to locate the ANN paradigm within this theoretical landscape, and to explore the potential of these systems for modelling and developing psychological theories of cognition. A more
specific aim is to consider the idea of a ‘neurocomputational approach to cognition’, which has been used
to characterise the ANN modelling enterprise in cognitive science, and to explore its viability in relation to
the classical information processing paradigm (that used to prevail in cognitive science). In order to
demarcate the scope of this study, these general aims are translated into the following more specific research
themes:

- The first theme explores the place of ANNs in the theoretical foundations of cognitive science. A
  standard interpretation of ANN models is that they are neural or ‘brain-like’ computational systems,
  and their contribution to cognitive science can therefore only be assessed if they are considered in
  relation to the mainstream, symbolic or ‘classical’ paradigm that has been the dominant
  computational research framework in cognitive science until recently. The relation between the
  classical and connectionist research paradigms is addressed in order to explore how ANNs fit in
  with the computational context that has developed under the classical rubric.

- A second theme is to explore the role of ANNs as representational systems, and to consider how
  the mode of representation used in connectionist approaches (i.e. typically distributed
  representational structures) differs from the symbolic scheme associated with classical approaches.
  These differences are considered in the context of ANN models of memory and conceptual
  structure, and the discussion is framed in terms of some of the theoretical and philosophical disputes
  that have arisen around the relevance of ANNs as computational models of cognition.

- The third theme is to consider the use of ANNs for modelling higher-level cognition. Much of the
  available research on ANNs concentrates on lower level cognitive processes such as memory and
  perception, but the impact of these models on higher level cognitive theorising is much more
  suspect, as Fodor and Pylyshyn (1988) have shown in a seminal article. The question of what
  exactly the contribution of ANNs are to the development of higher-level cognitive theories is
  therefore far from resolved. As will become clear, there are many difficulties that connectionist
  researchers have to overcome if these systems are to compete with classical systems in domains
  such as language and reasoning. On the other hand, these systems are much more compatible with
  cognitive linguistics conceptions of language, and can be used to model aspects of commonsense
  reasoning.

- A fourth theme is to consider the theoretical relevance of ANNs to the development of theories of
  cognition. PDP networks may generate interesting behaviour, but it is not clear that they do so by
  emulating the fundamental nature of human cognitive processes. In many cases the design decisions
  governing connectionist theory are determined by engineering needs so that there is a gap between
  connectionist technology and connectionist cognitive science. Connectionist networks can be
proven to be computationally powerful, but these proofs offer no meaningful constraints for designing cognitive models. The question about the theoretical significance of ANN models is therefore still unresolved in cognitive science.

Many alternative discussions of ANN models in psychology and cognitive science have been presented, but these tend to either ignore or underestimate the theoretical significance of the computational underpinnings of cognitive research. For example, two good expositions of connectionism written for cognitive researchers, Bechtel and Abrahamsen (1991) and Quinlan (1991) do not even mention the Turing Machine paradigm underlying contemporary cognitive research. ANN models are typically simulated on conventional digital computers, so that whatever the contribution is that ANN models bring to cognitive science, it cannot be as radical as some of the proponents of connectionism have suggested. Also these researchers downplay the associative principles that characterise the functioning of ANNs. I will present some arguments in later chapters to show that the contribution of ANN modelling so far to cognitive science is somewhat weaker than many connectionist researchers have argued, but that the approach has nevertheless an important role to play in cognitive science, even if it eventually survives only as an interesting methodological alternative to the development of standard, symbolic theories of cognition. On the other hand ANNs have forced cognitive researchers to rethink the role of computational approaches, the brain-mind connection, and particularly the role of mental representations in cognitive theory. These aspects associated with the connectionist approach may be sufficient to fuel its continued exploration by cognitive researchers.

1.3.1 The road ahead

The next chapter presents a historical overview of the development of connectionism, sketches the functioning of a typical ANN system, and reviews some important learning procedures used to train ANNs. The main aim of the chapter is to describe the generic connectionist architecture, which is typically feedforward, multilayered networks trained using backpropagation (Rumelhart, Hinton, & Williams, 1986). Models created from this generic architecture are sometimes argued to yield a different conception of cognition from that developed in the framework of classical cognitive science systems. Classical and connectionist models are information processing systems, but represent and process information differently. In order to be able to assess the similarities and differences between these approaches, it is necessary to highlight their computational properties. The chapter aims to isolate some of the main features associated with ANN models, so that such a comparison can be developed in later chapters.

Chapter 3 describes the classical computational approach that used to prevail prior to the resurgence of ANN modelling in the 1980s. In the chapter the foundations of the classical paradigm in Turing’s work on computational theory are described, and it is explained how this gave rise to the theory of functionalism and a hypothesis suggested by Newell and Simon (1976) in which the mind is considered as a ‘physical symbol
system’. The main aim of the chapter is to provide a systematic introduction to the classical framework so that the contribution of ANNs as information processing models of cognition can be interpreted within a proper computational context.

Chapter 4 focuses on ANN models of memory and conceptual structure. The possibility that connectionism might offer a radically new alternative for cognitive theorising has been resisted by some researchers who claim that it should rather be construed as a theory of how classical, symbolic theories are implemented in the neural system. On this view connectionist models are pitched at a different level of explanation (i.e. a level concerned with implementational issues) and do not compete with classical theories, but show how classical theories can be realised in neural-like hardware. One of the main strengths of the connectionist paradigm lies in its application to associative computing such as memory storage and retrieval. In the chapter, I consider distributed ANN models of memory and contrast these with the typical library filing metaphor of memory adopted in classical conceptions. Distributed representations require new ways of thinking about memory. In the file cabinet metaphor associated with classical approaches, discrete representations are thought of as being stored in unique locations that can be accessed independently. Even the metaphors, memory storage, search and retrieval evoke this familiar metaphor. With ANNs that use distributed representations, there is no discrete location for each representation, and accessing one representation entails accessing all, because representations are encoded in the same set of connection weights. Instead of search and retrieval, better metaphors involve similarity, and ‘resonance’.

Chapter 5 discusses the application of ANN modelling to higher-order cognitive functions such as language understanding. The chapter introduces Fodor and Pylyshyn’s (1988) argument that ANN models cannot successfully cope with the structural components of language. They identify a feature of human intelligence which they call “systematicity”, and contend that it poses serious difficulties for ANN models of language. They point out that there is an intimate semantic relationship between some sentences which are structurally and semantically very similar (e.g. between two sentences such as John sees Mary and Mary sees John). They then argue that the notion of ‘systematically related’ in the case of such sentences means that you don’t find subjects who know the meaning of one sentence but fail to grasp the meaning of the other (Fodor & Pylyshyn, 1988, p. 35). It is relevant to note that the two sentences are syntactically related, both exemplify a structure of the form [NP (VP) NP] where NP stands for noun phrase, and VP for verb phrase. So the systematicity claim boils down to the observation that in order to learn a language, speakers must induce the underlying syntactic structure and be able to parse novel sentences appropriately. The systematicity issue is a general challenge to connectionist modellers, but many responses to this challenge have ended in the construction of ANNs that incorporate classical-type facilities for coping with symbols. The open-ended question is whether ANNs can actually surpass, rather than just emulate, classical systems as language processors.
Chapter 6 considers the issue of whether ANN models can be regarded as theoretically relevant to cognitive science. First, researchers question the ability of such models to capture the right empirical generalizations. Second, such models are often exceedingly difficult to interpret, which mitigates against their usefulness as explanatory models of cognition. Third, in many cases the functional architecture of these networks is not completely specified. The success of computer simulations of psychological phenomena is often measured in the program's ability not only to make the same correct judgements as humans, but similar mistakes as well. Merely generating 'intelligent' behaviour does not guarantee a successful niche in cognitive science for an implemented theory, the important questions concern the contribution that ANN modelling actually makes to cognitive theory. An examination of the current state of research suggests a rather 'deflationary interpretation' of ANN models in terms of which they are likely to co-exist next to classical cognitive science models for some time to come. The chapter ends with a few final reflections on the future prospects of the connectionist orientation in cognitive science.
AN HISTORICAL SURVEY
OF NEURAL NETWORK MODELLING

In this chapter some of the historical forces and philosophical ideas that helped to shape the current generation of connectionist models are discussed. The connectionist paradigm has developed somewhat contiguously with the classical, symbolic computational framework in cognitive science. In a sense, connectionism is simply an extension of the classical approach from a serial to a parallel processing architecture, while most of the core assumptions of the approach are retained. Connectionism is also sometimes viewed as a different conception of mind, which has roots in an associative tradition, and stresses neural rather than symbolic computation (e.g. Smolensky, 1988). According to this latter view, connectionist explanations highlight analogical, associative and statistical procedures rather than strictly rule-based computation, and are pitched at a level closer to that of neural elements and processes.

In view of the historical link between ANN models and associative explanations of cognition, these models may seem to be subject to the known limitations of associationist approaches (Lachman, Lachman & Butterfield, 1979, p. 46). However, many researchers reject the tendency to equate ANNs with traditional associationism, pointing out that ANNs have powerful non-linear processing capabilities (see e.g. Horgan & Tienson, 1994). In fact, one of the so-called ‘allures’ of the connectionist paradigm in cognitive science is that these networks transcend the essentially linear capabilities of traditional associationist systems and can be used to implement complex, non-linear adaptive (i.e. self-organising) systems. There are good reasons to suppose that the human brain is a (nonlinear) dynamic system (Skarda & Freeman, 1987). Consequently, one of the motivations for the current surge in popularity of neural network techniques in psychology is that this technology enables researchers to investigate nonlinear processing mechanisms.

Given this background, the present chapter has three main objectives. The first is to describe the history of ideas that has shaped the neural network revolution in cognitive science, highlighting some of the milestones that have been attained in ANN modelling, and also mentioning some significant stumbling blocks that have, and may still to some extent, impede the development of sophisticated ANN learning procedures. The second objective is purely ‘remedial’, to debunk some of the mystique that accompanies connectionist theorising by showing that the models are for the most part based on fairly simple learning
rules that can be expressed in a few equations. Many ANNs are simply mathematical objects implementing a form of statistical computing. The notion that ANNs are really brainlike systems therefore needs to be treated with skepticism. The third is to point out that although some early ANNs were based on simple associative mechanisms, this is no longer true of many of the current architectures used in cognitive modelling, which have some nonlinear processing capabilities. Because these systems have nonlinear mechanisms, it would therefore be incorrect to dismiss ANNs as just associative machines, as some researchers have suggested (see e.g. Elman, Bates, Johnson, Karmiloff-Smith, Parisi, & Plunkett, 1996, pp. 47-50).

2.1 ROOTS OF CONNECTIONISM IN THE ASSOCIATIVE TRADITION

The operation of association involves the linkage of information with other information, and yields a mechanism for combining, representing and processing information. It is generally recognised that some aspects of human cognition, such as memory, function in an associative manner and for this reason the idea of associative processing has a long tradition in psychology. Associative processing mechanisms are also basic features of neural network computation. In fact, Hertz, Kroch and Palmer (1991, p. 11) start their discussion of the Hopfield model by observing that associative memory constitutes the "fruit fly" or the "Bohr atom" of research in neural computation, and suggest the following formal definition of association: "Store a set of patterns $\zeta$ in such a way that when presented with a new pattern $\xi_i$, the network responds by producing whichever of the stored patterns most closely resembles $\xi_i$." This definition captures an important aspect of associative systems, and hence also of connectionism, because it highlights the principle of computing the similarity between a given input pattern and another stored pattern. This computation of similarity (which typically involves continuous mathematics) differs from the essentially rule-based computations (which usually involves discrete mathematics) characteristic of the classical symbolic tradition.

ANNs are built up out of a number of processing elements, roughly modelled on neurons, that are joined together by connections analogous to the synapses connecting real neurons. The most common of these systems are pattern transformers which take an input pattern and transform it into an output pattern by modulating the numerical weights of connections and the activity patterns of the elements in the system. It is this ability to function as pattern associators that have attracted many psychologists to connectionism, because there is a substantial body of research suggesting that many aspects of human learning and memory processes are also based on some form of associative computation.
2.1.1. Associationism and the issue of human learning

The earliest version of the associative approach can be traced to Aristotle's views on human memory, articulated in the fourth century B.C. Aristotle made two bold claims about the operation of the memory system. He suggested firstly that the units of memory are sense images (i.e. sensory representations), and secondly that memories occur as a function of a linkage, an association, between sense images. He proposed several ways in which such linkages can be formed, namely by virtue of the similarity, contrast, and temporal contiguity of sense images (Sorabji 1972, p. 50). Aristotle's conception of the memory system is remarkably similar to contemporary views of associative memory. However, he did not elucidate the mechanism of associative memory in any detail, but posited an unexplained system of linkages as the source of memories (Anderson & Bower 1979, p. 16-17).

The associative movement as a theory of knowledge, an epistemology, was mainly worked out by the empiricist philosophers such as Berkeley, Hume and Mill (see Marx & Hillix 1979). On their account knowledge is stored in memory by a gradual accumulation of sensory information that is contiguous (either spatially or temporally) perceived, and such information becomes associated in the mind by virtue of a principle of causality that binds sensations together. In its early philosophical formulation, associationism is essentially an epistemology based on the fundamental tenet that knowledge is acquired by observing associations between spatially or temporally contiguous objects or events. In the empiricist tradition, sensory data is seen as the fundamental building blocks of knowledge and the process of association defines the way in which knowledge is acquired.

2.1.2 Thorndike's connectionism

The philosophical doctrine of associationism had a significant impact on the psychological theories of learning and memory developed by amongst others Watson, Thorndike, Hull, Tolman, and Ebbinghaus (Marx & Hillix, 1979, pp. 44-63). These theorists attempted to explain human learning and behaviour in terms of associative principles, but also developed an experimental approach to the study of memory and cognition, treating them as manifestations of associative mechanisms. It is also in this era that the notion connectionism was first introduced by Thorndike to describe a form of associative learning based on stimulus response contingencies. He presented his most significant contributions to psychology under the title “Writings from a Connectionist Psychology”, which he observes was written in the hope that future generations of scholars may “know something of connectionist psychology” (Thorndike, (1949, p.110).
Thorndike identified two laws mediating learning. The “Law of Exercise or Use or Frequency” states that the more a situation leads to certain response, the stronger becomes the tendency to emit the behaviour when the situation occurs in future. The “Law of Effect” states that what happens just after a behaviour had been emitted plays a role in the future production of the behaviour. These laws are clearly similar to the theory of operant conditioning developed by Skinner. Thorndike was interested in the neural mechanisms governing learning, and his theory of connectionism attempts to predict human behaviour by suggesting an associative organisation of information in the brain. The theory posits that learning involves the establishment of neural connections between the neural representations of stimuli and responses, and thus highlights the role of neural mechanisms in associative learning (see e.g. Marx & Hillix, 1979, pp. 55-56). Thorndike’s approach can be viewed as a precursor of the supervised ANN learning procedures, which will be discussed later in this chapter.

This neural focus is one of the main characteristics of the early psychological approaches to associative learning. In the 19th century the brain was pictured as a mass of neurons connected by synapses. The foundational role assigned to synaptic connections, and the conception of memory as modulations in synaptic transmissions prevailed since the time of Ramón y Cajal (1908), credited with the discovery of neurons, who suggested that learning is mediated by a process of cell outgrowth (see Valentine, 1989 for a good discussion). James (1890/1983) isolated both temporal and spatial contiguity as fundamental principles of learning and speculated about the neural basis for these processes. He suggested that the brain activity at any specific point is a function of the sum of the "tendencies" of all the other points connected to it, and further postulated that the strengths of the tendencies are proportionate to the frequency with which the points have been co-active, the intensity of the activity, and the relative absence of points emitting inhibitory activity. As Quinlan (1991, p. 3) points out, this position differs in important respects from the later behaviourist doctrine. James and the other 'early' connectionist researchers were reductionistic and attempted to isolate connections between psychology and neurology, whereas the behaviourists were concerned with overt behaviour and had little to say about the underlying neural mechanism. Almost contemporaneous with the work of James, Lashley put forward his hypothesis of "equipotentiality and mass action" which states in essence that the efficiency with which a complex cognitive function is performed will be reduced in proportion to the extent of a brain injury. His basic claim was that memory traces are not localised in the brain, and hence that the search for the elusive "engram" is doomed to fail. He suggested that the pattern and not the localisation of injury determines its functional impact on subsequent behaviour. Brains are resistant to damage, in terms of neural representation cognitive functions are not discrete, all-or-nothing systems and information is not stored anywhere in particular, but everywhere, because "...there are no special cells reserved for special memories" (Lashley, 1950, p. 455).
Emanating from these early approaches to associative learning are some of the fundamental concepts of contemporary connectionist views. There is a concern with the neurological substrata of cognition, an attempt to isolate neural processing mechanisms, an associative conception of the acquisition and storage of knowledge, and an emphasis on distributed representations at the neural level.

2.1.3 Behaviourism and the Rescorla-Wagner law

Unfortunately this preoccupation with the neural principles underlying association is absent from the later development of mainstream psychology during the behaviourist era. Behaviourism is really associationism in a slightly different guise. Its main theoretical constructs are stimulus and response dependencies, but behind these constructs lurks an approach to associative learning. In fact, the behaviourists presented two different theories of learning. The classical behaviourists conceptualised all learning as a process of habit formation, that is conditioning, in which associations are formed between co-occurring environmental stimuli, that is, stimulus-response associations. A second form of learning, called reinforcement learning has also been proposed.

In the case of reinforcement learning the assumption is that behaviour can be moulded by rewarding (i.e. giving positive feedback to) a spontaneously produced behaviour. The idea is that by selectively reinforcing desirable responses, an organism can be taught to approximate appropriate behaviours, and consequently that learning is mediated by some sort of behavioural shaping of responses. Although behaviourism ultimately failed as a theory of human learning, it has made some contribution to our understanding of the nature of associative processes. One of the most important results emanating from this approach is the Rescorla-Wagner theory of classical conditioning, which is now known to be formally identical to the delta rule (to be described later) used in the early models of connectionist learning (Anderson, 1995, p. 276). The principles of reinforcement learning have been successfully integrated into ANN learning using an algorithm called Temporal-Difference Learning (TDL). For instance, Garry Tesuaro trained a network called TD-Gammon to achieve a world class level of play in backgammon using TDL (Sutton & Barto, 1998, pp. 261-268).

The important point to note about behaviourist theorising is that they treated the mind as a black box. They were for the most part only interested in observable products of behaviour mechanisms, and not in the internal processes mediating the behaviour. As will become clear in this chapter, ANN modellers adopt a different approach because their focus is precisely on the cognitive and neural mechanisms that produce behaviour. Thus even though there are some similarities between the equations used in some
ANN learning procedures and the formalisation of behavioural performance emanating from behaviourist research, the theoretical aims of the two approaches are different.

2.1.4 Associative memory and neural computation

As can be gleaned from the short discussion of behaviourism in the previous section, the early psychological approaches to memory and learning embody a specific associationistic conception of mind. These associative approaches are characterised by the following four basic postulates (Bechtel & Abrahamsen, 1991, p. 102).

☐ mental representations or ideas become associated with one another through experience;
☐ these ideas are acquired on the basis of aspects such as spatial and temporal contiguity, similarity, and dissimilarity of experiences;
☐ complex ideas can be reduced to sets of simpler ideas so that simple additive rules can account for the development of knowledge;
☐ simple ideas are basically sensations.

These associative principles have been used to explain aspects of cognition such as concept formation, memory, and learning, but are now known to be inadequate when applied to the learning of higher level cognitive functions such as language. For instance in terms of associative learning principles the meaning of a sentence will be represented in memory as a chain of associations linking the verbal items on the basis of sequential dependencies. If so, the process of interpreting a sentence can be modelled by a Markov process or finite state grammar. Yet, it is now well known in formal language theory that finite state grammars are inadequate as a model of human language processing (see Chapter 3, Section 3.1.3.1). Moreover, although the early developments of the theory of associative learning have contributed to our understanding of the associative mechanism underlying human learning, they lack a clear processing (computational) account and disregard the internal, cognitive mechanisms of learning. Modern approaches to associative learning attempt to overcome these shortcomings by describing associative processes in terms of a more computationally orientated perspective. In this modern formulation, an associative memory is an extremely robust computational system, endowed with a few unique features, and most of the research is directed at elucidating (and formalising) the operation of these associative mechanisms. We now turn to consider some of the progress that has been achieved in understanding these associative mechanisms, and in building machines that perform associative computations.
2.2 TOWARDS A THEORY OF NEURAL COMPUTATION

Conceptually, ANN modelling implements a form of statistical computation (broadly know as “neural computation”, e.g. Hecht-Nielson, 1990), which in turn is inspired by the operation of the brain. It draws from a distinctive metaphor for cognitive research, founded on ideas of massively parallel and distributed processing of the brain itself. Furthermore, in contrast to the abstract approach often adopted in traditional computational models of cognitive systems (more about this in the next chapter), connectionism is concerned with issues of how cognition is realised in the wetware of the brain, and connectionist researchers typically try to develop models that emulate the functioning of the neural system in some general sense.

2.2.1 Biological neurons

It is helpful to review a few basic aspects of the brain's functioning because artificial neural networks are said to be “inspired” by the brain (Rumelhart, 1992), and attempt to model the brain albeit at a fairly high level of abstraction.

The human brain has a very large population of neurons (approximately $10^{10}$), each connected to many other neurons, yielding a dense web of interconnections. The operation of single neurons is complicated and not fully understood at the microscopic level, although the basic ideas are relatively clear (see e.g. Thompson, 2000). Essentially a neuron merely accepts inputs from other neurons, sums these inputs in some way and either fires (if the summed inputs exceed a given threshold), or remains inactive. Despite the apparently simple operation of single neurons, complex information-processing abilities can emerge from collections of neurons working in concert. The major components of the natural neurons are shown in Figure 2.1 below:

![Figure 2.1 A biological neuron](image-url)
Each neuron consists of a cell body, the *soma* which has two types of connections projecting outwards to other neurons. The first consists of branch-like irregularly shaped filaments called *dendrites*. The dendrites are essentially afferent connections and serve to accept input from other neurons. Simplifying somewhat, we can say that the dendrites perform a summation function on the input arriving at the cell body. The dendrites converge at the soma and convey a single summed neural pulse to the cell. The second connection consists of a long filament called the *axon*, situated orthogonally to the dendrite connections on the cell body. The axon functions as an efferent connection, it is an output channel relaying neural impulses to other neurons. Its operation can be envisaged as the computation of a non-linear threshold function, which results in the emission of a voltage pulse that lasts about $10^{-3}$s, called the *axon potential*. The pulse is produced when the resting potential inside the soma rises above a certain critical level, which serves as a threshold. The axon potential is in fact a series of rapid voltage spikes. The axon terminates in a specialised, bulb-like contact called a *synapse* which serves as a coupling between the axon and the dendrite of another cell. There is no direct physical contact at the junction between the axon and its dendrite, rather the connection is achieved by virtue of a temporary chemical channel. The synapse contains small vesicles carrying neurotransmitters, which are released when the potential within the synapse is raised sufficiently by the action potential. The neurotransmitters diffuse across the gap and chemically activate gates on the dendrites. Charged ions enter at the open gates and propagate along the dendrite filament to the cell body of the receiving soma. The influx of ions alters the dendritic potential and produces a voltage impulse (Thompson, 2000, pp. 2-5).

At the synaptic junction, the number of dendrite gates that open depends on the number of neurotransmitters released. In addition, synapses can either excite or inhibit the dendrites they affect. Inhibition corresponds to altering the local potential of the dendrite in a negative direction. Learning in biological systems is thought to result from changes in the cell couplings at the synaptic junction. Such changes enhance the synapse's ability to transmit neurotransmitters which, in turn, has the effect of increasing the permeability of the membrane on the post-synaptic side, thus facilitating the conduction of charged ions by the receiving dendrite. An influx of positive ions into the post-synaptic cell will tend to depolarise the resting potential producing an excitatory effect on impulse conduction. In contrast, the release of negative ions into the post-synaptic cell causes hyperpolarisation, and produces an inhibitory effect.

The account given above highlights the function of single neurons. As this explanation illustrates such neurons are essentially processing elements which serve to transmit impulses across the neural system, so they can be conceptualised as the basic computational units of the brain. The computational
mechanisms of artificial neural nets draw from the operation of such natural neurons, but they exploit a fairly basic conception of the functional features of a neuron. The main focus of perhaps the majority of ANN modellers is rather on the development of techniques for constructing networks of such ‘formal’ neurons, than to capture all the properties of single natural neurons. This concern with larger scale brain operations has traditionally been the domain of systems neuroscience, and it is here that neural network theory has left its mark.

2.2.2 From natural to formal neurons

As already noted, neural network modelling is a research programme directed at creating formal models of brain functioning. This interest in modelling the operation of the brain derives from two related pursuits. It is pursued both from the perspective of science, because the brain is the seat of human intelligence and thus by uncovering the working of the brain it is hoped that we may gain insight into the mechanism of mind. However, understanding the operation of the brain is also important from an engineering point of view because the brain is a working model of an intelligent system, and the knowledge gained from the study of the brain might eventually help to advance the design and development of intelligent machines. Both pursuits have had an impact on the evolution of the neural network paradigm, and we shall return in the last chapter to consider their contributions in more detail.

For now, the first task is to develop a better understanding of what a neural network is, and here it is useful to start by first taking a closer look at its basic processing unit, the ‘formal neuron’. Figure 2.2 below illustrates a generic neural processing neuron. It comprises:

- A set of synapses or connections each represented by a weight or connecting strength. In the figure, \( w_i \) indicates the connection weight (i.e. the synapse) stemming from neuron \( i \). The synapses are the receptors of the neuron and accept input signals from the other neurons and from the environment. These input signals mediate between the neuron and the content domain within which it is required to operate. Viewed in terms of natural neurons we may say that if the weight is positive, the associated synaptic connection is excitatory and if it is negative, the synaptic connection is inhibitory. The input signals can be envisaged as a vector, \( I \), and the connecting weights as another vector \( w \).

- An internal bias that serves to increase the neuron’s resistance to the effect of input signals impinging on it. This bias is a formal analogue of the resting potential of a natural neuron and functions as a threshold so that a given neuron only fires if its internal threshold is exceeded.
An adder (or summing junction) which sums the various input signals weighted by their associated synaptic connections weights. Formally this task entails combining two vectors (i.e. the inputs and weights), typically yielding their inner product $\mathbf{I}^\mathbf{w}$ (although some learning algorithms such as the Hebbian procedure - which is discussed a little further down- require computation of the outer product $\mathbf{Iw}^T$).

An output function (or activation function) which modulates the output of a neuron. The output function is often referred to as a squashing function because it has the effect of limiting the amplitude range of a neuron to some finite value. This amplitude range is typically taken to be one of the closed unit intervals $[0,1]$ or $[-1,1]$.

![Diagram](image)

**Figure 2.2** The operation of the generic, formal neuron

As the figure above shows, the operation of such a typical connectionist neuron consists in first summing the input signals. Mathematically, this summing operation can be expressed as:

$$Net_j = \sum_{i=1} w_{ij} I_i$$  \hspace{1cm} (2.1)
where Net\textsubscript{j} is the result of the linear combiner (indicated by the summing junction in the figure above), \(I_i\) is the set of input signals the neuron receives, and \(w_{ij}\) is the weight (a numerical value) of the synapse connecting neurons \(i\) and \(j\). For simplicity of notation, we can assume that input signals are simply the output activation of a unit in the previous layer, and denote activation as \(y\). The signal arriving at unit \(j\) from unit \(i\) can then be written as \(y_i\), and the activation leaving the unit would be \(y_j\). Upon receiving weighted inputs, an output activation for the neuron is determined as a function of the net input and the bias as shown below:

\[
y_j = f(\text{Net}_j - \theta_j)
\]

(2.2)

Here \(\theta_j\) denotes the bias or threshold of the particular neuron concerned, that is, neuron \(j\), and \(f\) is an output function. It is mathematically ugly to include the term \(-\theta\) in the previous equation. However, if we assume that the unit’s bias has a constant input value of -1 and a weight equal to 0, the simpler expression (relative to Equation 2.2) shown below obtains. This formulation will be assumed throughout the rest of this chapter.

\[
y_j = f(\text{Net}_j)
\]

where \(\text{Net}_j = \sum_{i=0}^{\text{Net}} w_{ij} I_i\)

(2.3)

In (2.3) the summation subscript decreases to 0 to cater for the ‘extra’ addition of the bias input. In the figure, \(f\), the output function is a squashing function, which compresses the output between two extremes (e.g. 1 and 0). Notice that the function illustrated in the figure is nonlinear (it is not represented as a straight line); such non-linearities are typically used in the multilayer ANN models that will be introduced later in this chapter. As already mentioned, the ability to perform nonlinear processing is a distinctive aspect of modern ANNs, and something which sets them apart from older associative mechanisms.

Variations of the generic formal neuron described above are pervasively used in ANN modelling, but it should be immediately obvious that it entails an extremely simplified depiction of the functioning of natural neurons. There are a variety of different neurons in the brain, but in ANN models this diversity is simplified to a single ‘generic’ model. Also an accurate description of how the membrane potential spreads, requires complex equations to account for the spatial and temporal effects of the neural impulse’s propagation along its connections (see Anderson, 1995, pp. 27-31). Moreover, the operation of neurons are governed by a complex interaction between genetic and biochemical processes, which are ignored in the ANN neuron shown above. There is software available that provide a reasonably detailed
simulation of the operation of real neurons (e.g. Hines & Carnevale, 1997), but the models typically used in ANNs provide only a broad functional description of neurons as integrators of information.

2.3 SOME EARLY NETWORK MODELS

Much of the historical development of connectionism consisted in work on three interrelated aspects, involving:

- the computational elements, the formal neurons, that perform the information processing operations in the network;
- the neural network architectures in terms of which the neurons communicate with one another;
- and
- the learning algorithms that enable the neural networks to acquire the relevant information processing capabilities through exposure to pertinent data.

During the course of these developments, these systems evolved from an initial experimentation with very simple neuron-like processing units to gradually more complicated collections of ‘neurons’, and eventually culminating in complex neural network architectures endowed with a learning capacity. However, neural network research has become such a vast and complicated topic, and it has roots in so many disciplines (e.g. neuroscience, cognitive science, psychology, mathematics, computer science, physics, statistics, and engineering) that it is becoming increasingly difficult to present a concise overview of the area (which as already noted, is now best treated in book length presentations). As a result, the discussion below is somewhat superficial and highlights only some of the main stages in the history of connectionism. Also, in what follows the focus is mainly on the development stages leading to the publication of the backpropagation learning procedure, because this algorithm, and variations based on it, are predominantly used in most ANN models used in cognitive sciences. A thorough survey of the history of connectionism is presented in texts such as Hagan, Demuth and Beale (1995), Haykin (1999), and Schalkoff (1997).

2.3.1 The McCulloch and Pitts’ neuron

The idea of computing machines that incorporate neural features had its onset in the cybernetic era. Norbert Wiener and his colleagues formulated the idea that biological entities can be conceptualised from an engineering and mathematical perspective as feedback mechanisms. At the same time McCulloch (a
neurophysiologist) and Pitts (a mathematician) published the first formal treatment of an artificial neural network (McCulloch & Pitts, 1943). They tried to show that primitive logical operations can be executed by formal neurons, and that complex logical processes might emerge from the activity of collections of such neurons. The resultant neuron, known as a McCulloch and Pitts (henceforth M&P) neuron, formalises the idea of a neuron as a computational device. Their goal was to define a neuron that can compute the truth value of a set of logical propositions, by outputting either 1 (if the proposition is true) or 0 (if the proposition is false). The M&P neuron functions as a threshold logic unit (TLU). It accepts a set of binary valued signals as inputs, which are then summed by calculating the inner product of the signals and the weights (also binary valued) on the connection links to the neuron. If the resultant value exceeds the threshold value, a value of 1 is transmitted as output, else 0 is transmitted. The unit’s output function is linear, defined by a binary logic.

Formally, the neuron computes its input activation by performing a simple summation of the input signal:

\[ Net = \sum w \cdot y_i \]  

(2.4)

It then computes an output activation, based on a threshold function, which determines the output generated by the neuron based on a simple binary decision criterion such as: If \( Net \leq 0 \) then \( y = 0 \) and if \( Net > 0 \) then \( y = 1 \), where 0 is the threshold for the unit.

Although simple in conception the M&P neuron is a computationally powerful device. McCulloch and Pitts proved that a synchronous assembly of such units could in principle function as a computational device provided suitable weights can be found, and that it would be equivalent in computational power to an ordinary digital computer. Because the unit processes its input in terms of a basic binary logic (i.e. its output is restricted to either 1 or 0), the Boolean functions AND, OR, and NOT can be represented by units endowed with suitable weights and thresholds. For example, the Boolean function AND can be modelled by setting the threshold value of the neuron at 2, so that it will only output ‘true’ if the sum of both input connections to the neuron have values of ‘1’. On this basis they were able to show that any proposition or statement expressible in propositional logic can be represented by a network of simple processing units. McCulloch and Pitts (1943, p. 131) concluded that:

“To psychology, however defined, specification of the net would contribute all that could be achieved in that field - even if the analysis were pushed to the ultimate psychic units or "psychons", for a psychon can be no less than the activity of a single neuron.”
The term ‘psychon’ in the citation was later refined in McCulloch (1965/1950, p. 8) and refers to the "least kind of psychic event" which, in turn, essentially boils down to a primitive proposition that is either true or false. Although McCulloch and Pitt’s work is certainly important because it serves to highlight the expressiveness of ANNs, there are a number of drawbacks with their proposals when evaluated in terms of biological criteria:

- Real cells can perform a nonlinear summation of their input, whereas the M&P neuron computes a linear summation of inputs;
- A real neuron produces a sequence of pulses, while the M&P neuron generates a single output value.

A more fundamental shortcoming is that McCulloch and Pitts did not describe a learning algorithm for their neurons, nor did they succeed in assembling a network of such neurons to actually perform intelligent tasks (although they did conjecture that it would be possible to construct such nets).

### 2.3.2 Hebbian learning

McCulloch and Pitts were mainly concerned with a formal analysis of the computational properties of networks of units. In contrast, the psychologist Donald Hebb concentrated on the neural mechanisms underlying human learning, but did not develop his theories in formal detail (as McCulloch & Pitts, 1943 did). He nevertheless explored the physiological basis of mental processes and suggested a form of learning at the cellular level that is now generally known as Hebbian learning. Basically Hebb (1949, p. 62) postulated that the excitation of a cell A, located in close proximity to another cell B, modulates the activity of B. More specifically, he claimed that when there is a persistent activation of B so that B’s firing efficiency is augmented, some "growth process or metabolic change" occurs which produces a concomitant effect on the ease with which A becomes excitated. This statement is known as Hebb’s rule, and can be reformulated in the following way:

- If two neurons on either side of a synaptic connection are activated simultaneously (i.e. in synchrony), then the strength of that synapse is selectively increased; and
- If two neurons on either side of a synapse are activated asynchronously, then the synapse is selectively weakened or eliminated.

Although Hebb did not do it himself, the rule represents a form of learning pitched at the neural level
which can be described in the language of neural network modelling. In Hebbian learning, weights between learning units (neurons) are adjusted in an attempt to arrive at an optimal representation of the relationship between the units. Units which tend to be positive or negative at the same time will have strong positive weights while those whose activity is asynchronous will have strong negative weights. Units which are uncorrelated will have weights near zero. The general formula for this type of learning is:

\[ w_{ij}(t+1) = w_{ij}(t) + \eta y_j(t) y_i(t) \]  \hspace{1cm} (2.5)

In this equation \( y_i \) and \( y_j \) represent the output values of neurons \( i \) and \( j \), connected by the synapse \( w \), and \( \eta \) is a parameter, set between 0 and 1, which denotes the learning rate. An important property of this rule is that it produces local learning in that the change in synaptic weights depends only on the activities of the two neurons linked by the weight.

Hebb’s main concern was to explain how networks of associations, which he called “cell assemblies” are formed. Hebb suggested that structural changes develop through reverberatory activation because certain primitive cell assemblies act as feature detectors and whole cell assemblies develop at higher levels. For example, the idea is that in perceptual learning, visual inputs are decomposable into primitive elements through the operation of innate perceptual analysers, and that simple cell assemblies are built up around the neural stimulation resulting from the activity of such analysers. Hebb postulated that the temporal relationship of activity in the various sub-assemblies, which constitutes a phase sequence, eventually culminates in the formation of neural networks for higher-level perception (see Quinlan, 1991 for a discussion).

A drawback of Hebb’s learning rule is that he described cellular learning purely in terms of excitatory activity, and did not allow for the formation of inhibitory connections between cells. This omission may be ascribable to parsimony considerations in that Hebb focussed on only one process because the introduction of a second process (i.e. inhibition) would necessarily involve complex interactions and there was no obvious way in which the integration of the two processes could be described at the time (Quinlan, 1991, p. 7). Instead, Hebb relied exclusively on excitation, treating inhibition either as a lack of excitation or as a form of neuronal refractoriness. There are some other problem with Hebb’s account. For instance, his approach is based on the idea that the pre-synaptic and post-synaptic cells are both fully active during learning, but recent evidence belies this hypothesis (e.g. Alkon, 1989). Nevertheless Hebb’s learning rule is still used in some recent connectionist systems (e.g. Linsker, 1986) particularly
because of its simplicity and biological plausibility. One problem with Hebb’s simple learning rule is that connections weights between neurons can become infinitely large as learning continues, but this issue has been addressed in various ways in contemporary networks. A rather obvious solution to this problem is to normalise the weight updating so that the weights remain bounded (but see Oja, 1982 for a more ingenious solution).

Despite some problems associated with Hebb’s (1949) rather simple, intuitive formulation of a mechanism of neural learning, it remains one of the historical milestones in the history of neural nets. There is a direct relationship between Hebb’s principle and subsequent neural network learning algorithms such as Kohonen’s self-organising systems (which will be considered later in this chapter), and adapted versions of Hebbian learning continue to figure even in recent neural net models (e.g. O’Reilly, 1996).

2.3.3 Selridge’s pandemonium

Oliver Selridge published Pandemonium in 1958, a creative and unusual information processing system. The rather suggestive name is indicative of both the processing units themselves as well as the manner in which they perform their operations. It is perhaps best understood as an interesting variation on the homoeculi theme, where instead of little men, Selridge (1958) postulated an hierarchy of cognitive demons that performed the cognitive operations. The system is extremely parallel in its construction and execution, and comprised of many simple (but not simplistic) units. At the bottom the data demons serve merely to store and pass on the data. At the next level the computational demons or subdemons perform certain more or less complicated computations on the data and pass the results of these up to the next level, the cognitive demons who weigh the evidence, as it were. Each cognitive demon computes a ‘shriek’, and from all the shrieks the highest level demon of all, the decision demon, merely selects the ‘loudest’, (Selridge 1958, reprinted in Anderson and Rosenfeld 1988, p. 118).

It is important to emphasize the parallelism present in Selridge’s model. Each demon in the array is computing its function - the computational demons are filtering out the recognisable features from the noise, the cognitive demons compute a shriek based on the strength of the evidence they have been provided with, and they do this all at the same time. It is only after all the shrieks are voiced that the decision demon earns its keep.

The initial structure of the Pandemonium is determined by processing task, but at the computational level
it is modified by two different learning procedures. The first procedure entails adjusting the connection weights linking the cognitive and the computational demons using a supervised learning technique. In this learning technique the weights are trained using a hill-climbing procedure to optimise the behaviour of the network. The second learning mechanism kicks in after the first has run long enough to achieve approximately optimal behaviour. It selects the computational demons with the highest worth (i.e. contribute the most to the network’s optimal performance), eliminates those with low worth and then generates new ones for those that have been eliminated. The creation of new demons is achieved by either mutating a demon or by conjoining two successful demons into a single entity. As this discussion illustrates, there is clearly considerable similarity between Selfridge’s (1958) machine and subsequent use of genetic algorithms in machine learning (see Goldberg, 1989). It is also interesting to note that genetic algorithms are now often combined with ANN techniques as illustrated by Biocomp Systems’s (1997) “Neural Genetic Optimizer”. In this approach ANNs are used to learn the input-output mapping in the data, and genetic algorithms are used to grow and experiment with different network structures.

2.3.4 Rosenblatt and the Perceptron

Hebb’s work laid the foundation for the study of learning at the neural level, and coincides with some other major advances in neural network modelling. Approaches with a more computational orientation occurred during the same period, 1940-1950. During this time, Marvin Minsky (later one of the main critics of the neural network research programme) developed the first neurocomputer, which he called the “Snark”. It realised Minsky’s technical objective of creating a neural-like processing system, but did not perform any interesting information-processing functions. The system consisted of forty electronic units interconnected by a network of links, each endowed with an adjustable probability of receiving and transmitting activation signals. It learned by means of a reinforcement process in which each positive and negative judgement about the machine’s behaviour was translated into a small change (of corresponding magnitude and sign) in the probabilities associated with the most recently active connections. Other similar systems gave birth to the specialist field of adaptive control, which was an important force in the cybernetic era (Levine 1983).

A dramatic breakthrough came in the early 1960s with the publication of Rosenblatt’s (1962) book which introduced a neurally inspired computational machine called a perceptron. Rosenblatt investigated various classes of such machines, but they were typically formulated as a layered network with continuous rather than binary processing elements and containing three types of units, i.e. sensory (S), associative (A), and response (R) units. The terminology is deliberately cognitive in inspiration because,
as the name implies, Rosenblatt’s objective was to create a psychologically plausible computational model of the visual processing system. The sensory units functioned as input devices which transmitted incoming information to associative units, which performed the primary processing functions. The response units emitted the perceptron’s output on a given processing task which typically consisted in categorising stimuli in terms of a predetermined set of category types. In more contemporary connectionist language, the perceptron can be described as a strictly feedforward processor of information with one way connections from $S$ to $A$, and from $A$ to $R$, and with only one or more $R$ units each connected to all the $A$ units. The $S$ and the $A$ units were only partially connected, and the weights of all the connections between $S$ units and $A$ units were fixed so that they did not change with time.

An important characteristic of the perceptron was that it could learn. Learning occurred by training the perceptron on a data set consisting of input values and output values of a function that the machine was required to approximate, and this function approximation was achieved by means of a relatively simple error correcting procedure (which will be described below). The perceptron’s neuron is similar to the M&P neuron, but it accepts a vector of inputs, rather than just binary inputs, and performs a weighted sum of the input, by calculating a simple inner product. Thus for unit $j$ we have:

$$Net_j = \sum_i y_i w_{ij}$$

where $Net$ is as defined previously, $y_i$ is the activation coming from input unit $i$ and $w_{ij}$ is the weight connecting unit $i$ and unit $j$. The output is determined by testing whether the sum of the inputs, computed as shown above, exceeds or falls below a given threshold:

if $Net_j > \theta$ then $y_j = 1$, and if $Net_j \leq \theta$ then $y_j = 0$

Here $y$ denotes the output value of the unit in question, and $\theta$ its threshold or bias. Rosenblatt’s significant contribution lay in incorporating a mechanism for updating weights. The perceptron convergence rule (Rosenblatt, 1958) attempts to reduce the discrepancy between the actual and desired activity of the output units by adjusting both the threshold of the output unit, and the connection from input to output. If the activity level is too low, then the threshold value is decreased and the connection weights are increased. If the activity level is too high, then the threshold value is increased and the connection weights are decreased. The learning procedure entails the calculation of the difference between the target or ‘desired’ output and the network’s actual output, and using this difference as a basis for adjusting the weights and the threshold of the output unit. This approach entails updating the weights
in the network by first computing the difference between the target and the actual value emitted by the perceptron. The difference can be written as

\[ d_j = (t_j - y_j) \]  \hspace{1cm} (2.7)

where \( d_j \) denotes the result of the difference between these two values for unit \( j \) after presentation of a particular pattern. For a perceptron that uses just 0 or 1 as inputs, the result of 2.7 will be zero if the inputs are the same and either +1 or -1 if they are different. Rosenblatt (1962) defined a number of different learning rules, but in the simplest one, a constant is added or subtracted from the weights during learning. It can be stated as:

\[ w_{ij}(t+1) = w_{ij}(t) + \eta d_j l_i \]  \hspace{1cm} (2.8)

where

\[ \eta = \text{a small constant defining the learning rate} \]

\[ d_j = 1 \text{ if } t_j > y_j; \ 0 \text{ if } t_j = y_j; \ -1 \text{ if } t_j < y_j \]

\[ l_i = 1 \text{ or } 0, \text{ the value of the input unit } i. \]

Notice that in terms of (2.8), a weight is changed only if its input is active (i.e. has a non-zero value), and if \( t_j \) and \( y_j \) are different. The parameter \( \eta \) controls learning in that it is added to the weight if the target value is higher than the actual value, and subtracted if the converse holds. The value of \( \eta \) is usually set below 1, and determines the learning rate in that learning is faster when it has a large value than when it is set to a small value.

In addition to inventing a learning algorithm for the perceptron, Rosenblatt also proved that the perceptron algorithm is capable of learning any \textit{linearly separable} function (this restriction will be described below), given enough training examples. This important result is known as the "Perceptron Convergence Theorem" (Rosenblatt, 1958). The theorem postulated that given a general type of categorisation problem, the network is guaranteed to learn from some finite set of examples presented to it. This remarkable result ushered in a flurry of activity in engineering and computer science in which researchers tried to develop perceptron-like systems to learn complex functions in pattern recognition, such as handwriting and speech recognition. Unfortunately these applications proved beyond the capabilities of this system, because it soon became evident that the perceptron suffered from some serious limitations (the source of the difficulty will be disclosed a little later in this chapter).
Almost contemporaneous with Rosenblatt's work on the perceptron, Widrow and Hoff (1960) developed a neural network known as the Adaline in 1960. Their network was an outgrowth of digital signal processing studies and was the first network to learn by an iterative procedure. The architecture of the neuron used in the network is similar to that of Rosenblatt's perceptron, and likewise computes a weighted sum of all its inputs, after which a threshold function is applied so that a +1 or a -1 signal is transmitted as output. There is one input and one modifiable synapse for every element in the input array (i.e. the pattern presented as input to the network). The net input activation to a given output unit \( j \) is:

\[
Net_j = \sum y_i w_{ji}
\]  

(2.9)

where as before \( y_i \) denotes the activation coming from input unit \( i \).

The Adaline is trained by presenting the training set to the network, one pattern at a time. With each input pattern, the correct or expected or target values of the output pattern is also presented. The network generates an output pattern which is compared to the target pattern. The algorithm then works as follows: If the network managed to generate a correct output value the weights are left unchanged. If it generates an incorrect output value, the weights are modified according to a rule known as the 'delta rule', described in Widrow and Hoff (1960). The rule involves changing the weights in proportion to the distance between the actual output and expected output values. The distance is determined by subtracting the network's actual output from the expected output. This error value is then multiplied by a learning constant and by the input pattern vector to compute each specific weight change. The goal of error-correcting neural network learning procedures is typically to make adjustments to the weights in the network in order to minimise the error in the output produced by the network. In the Adaline gradient descent is set to minimise the error in the network's output using the delta rule:

\[
\Delta w_{ji} = \eta y_i \delta_j
\]  

(2.10)

In this expression \( \eta \) is a learning parameter, \( y_i \) is the signal received from the unit in the input layer, and \( \delta_j \) represents the difference (delta) between the target and actual value of unit \( j \). A derivation of this rule for adjusting the weights in the network is presented in Rumelhart, Hinton and Williams (1986), and is also sketched in Appendix A. A distinctive aspect of the delta rule equation is that it yields a learning procedure that allows the pattern of observations to be entered sequentially, and it therefore captures the
on-going characteristics associated with adaptive processing. This rule is also known as the “least mean squared” error (LMS) and, as noted above, implements a form of gradient descent because it attempts to minimise the error in output pattern by descending down the gradient in the error landscape.

2.3.6 Minsky and Papert’s critique: The XOR problem

Although Widrow and Hoff’s work was influential, the most significant development in connectionist research during this era was Rosenblatt’s perceptron. Unfortunately it soon became apparent that perceptrons are severely limited in their representational capacity, because the set of perceptrons actually constructed by Rosenblatt in the course of his experimentation could only learn linearly separable functions. To understand the concept of linear separability, a little geometric imagination is needed. Consider two classes of patterns, which we can denote by Class 0 and Class 1 respectively. We can think of a pattern simply as a point in n-dimensional space, with the coordinates of the point representing attributes associated with the object to be classified such as weight, texture, frequency. The two classes are said to be linearly separable if they can be separated from one another by means of a simple linear dimensional hyperplane. In 2-dimensional space a hyperplane is a line, in 3-dimensional space it is an ordinary plane, and in n-dimensional space it is an (n-1) surface. To learn a particular function, a perceptron had to determine a hyperplane such that all elements of Class 0 are on one side of the hyperplane and all elements of Class 1 on the other side. Classes that have this property are said to be linearly separable, but classes which do not have this property are “not linearly separable”. A perceptron outputs a 1 only if \( \mathbf{W} \cdot \mathbf{I} > 0 \), with \( \mathbf{W} \) denoting the weight matrix and \( \mathbf{I} \) the input vector. This means that the entire input space is divided in two along a boundary defined by \( \mathbf{W} \cdot \mathbf{I} = 0 \), that is a plane in the input space with coefficients given by the weights. With \( n \) inputs the input space is \( n \)-dimensional. To illustrate, consider the two figures below which show the inclusive and the exclusive OR functions.

![Diagram](image)

**Figure 2.3** A straight line can be drawn to separate the dark circles and the light circles in the case of the
inclusive or-function, but not in the case of the XOR function

Each function can be represented as a two-dimensional plot defined by the two input values as shown by Figure 2.3. In the case of inclusive or, the function only yields an output of false (i.e. it generates a "0") if both inputs have the value of "0". In the case of the XOR function, it is only true if either one of the inputs is true (i.e. has a value of 1) and the other false (i.e. has a value of 0).

We let black dots indicate a point in the input space where the value of the function is 1, whereas white dots indicate a point where the value is 0. A perceptron can only represent a function if a line can be drawn that separates all the white dots from the black dots. As shown in the figure, no line can be drawn that separates the white and the black dots in the case of the XOR function, and it therefore follows that the perceptron cannot learn this XOR function. Minsky and Papert (1969) showed that this restriction applies to all functions which are not linearly separable. They also pointed out that the perceptron suffers from a number of further shortcomings, such as that it is incapable of separating connected from unconnected figures because it cannot compute connectedness as a category, which implies that a perceptron cannot be taught to distinguish between connected and unconnected figures in any practical sense. They furthermore claimed that while the learning procedure can be used to categorise simple patterns of data, it cannot scale up to deal with much more complicated patterns. Rosenblatt realised that more complicated perceptrons, incorporating a hidden layer, would be capable of more complex learning (see Hecht-Nielsen, 1990, p. 15-19). He even experimented with the idea of such multilayered perceptrons, but was unable to devise a learning algorithm for them. Minsky and Papert somewhat cavalierly dismissed the latter category of perceptrons as "not contributing anything of significant value to the perceptron" (Minsky & Papert, 1969).

Hecht-Nielsen (1990, p. 17) suggests that Minsky and Papert may have had a hidden agenda and that they were mainly concerned with diverting research funds allocated by the defence administrations (the main funding agency for AI research at the time) away from the perceptron programme to symbolic AI research which was represented, inter alia, by Minsky and Papert themselves at the Massachusetts Institute of Technology's AI laboratory. However, Rosenblatt himself was partly to blame for the fact that the perceptron enterprise turned sour, as can be gleaned from his own account of the problems he faced:

"There seemed to have been at least three main reasons for the negative reactions to the program. First was the admitted lack of mathematical rigor in preliminary reports. Second, was the handling of the first public announcement of the program in 1958 by the popular press, which fell to the task with all the exuberance and sense of direction of a pack of happy bloodhounds. Such
handles as "Frankenstein Monster Designed by Navy Robot That Thinks" were hardly designed to inspire scientific confidence. Third and perhaps most significant, there has been a failure to comprehend the difference in motivation between the perceptron and the various engineering projects concerned with automatic pattern recognition, "artificial intelligence," and advanced computers." (Rosenblatt, 1962, preface).

Still, Minsky and Papert's (1969) detailed, rigorously mathematical, and systematic analysis of the perceptron and the highly esteemed reputations of the two authors, coupled with their negative depiction of the perceptron's potential to simulate human perceptual processes, had a dampening effect on further work on the perceptron. Rosenblatt died shortly afterwards and without his inspiration, the research programme began to dwindle.

2.4 THE POST-PERCEPTRON ERA

Subsequent to Minsky and Papert's now famous critique of the perceptron, most research activity in AI and the cognitive sciences moved to symbolic computation. Not much explicit neural network research was conducted, but a body of research persisted under the rubrics of adaptive signal processing and pattern recognition (Hecht-Nielsen, 1990, p. 18). In addition, a few staunch supporters (e.g. Stephen Grossberg at Boston University and James Anderson at Brown University) continued to experiment with neural network modelling techniques.

2.4.1 Hopfield's attractor networks

Hopfield was a highly respected physicist with important institutional connections at CalTech and Bells Communications Laboratories. In his hands ANNs became legitimate, whereas before, most developments in networks had been the province of somewhat suspect psychologists and neurobiologists, or by those removed from the hot centres of scientific activity (Anderson and Rosenfeld 1988, p. 457, cited in Bechtel and Abrahamsen 1991). Hopfield's work therefore did much to spread the field within the physics community. His most significant contribution was to suggest a close connection between physical systems and ANNs. In particular he pointed out that simple networks with symmetrical connections behaved as if they were minimising a quantity that acted like the energy of a physical system. This gave a firm theoretical foundation for a type of nonlinear, crosscoupled network that computes by settling into a stable state.

The Hopfield network is a symmetric network in which each unit is connected to all others. Because the
network is symmetric, the weights on the net are the same in both directions, so that \( w_{ij} = w_{ji} \). In this network there are no obvious input or output connections, and all the units (i.e. nodes in the network) are equivalent. Each unit in the net is assigned a threshold and a step function. The units compute the weighted sum of the inputs minus the threshold value, passing the result through the step function to obtain the output value. Inputs are applied to all the units at once, and involve a set of starting values presented to the network. Upon presentation of the input, the network cycles through a succession of states until it converges upon a stable solution where the weights no longer change. An easy way to imagine the operation of the network is to consider that all the units affect one another due to the interconnected architecture of the model. Initially the network will be unstable because there is a competition between the activation values on the units, some units will try to make a given unit turn on, while other units will try to make it turn off. As the network moves through a succession of states it will gradually become stable as the units vying for attention begin to reach a compromise. The final or stable state of the network represents the ‘best compromise’ or solution to the initial activation of the units.

Hopfield used an energy function as a tool for designing recurrent networks, and for understanding the dynamic behaviour of these networks. His approach focussed on the principle of storing information as dynamically stable attractors and popularised the use of such recurrent networks for developing computational models of associative memory and for solving combinatorial optimisation problems. A Hopfield network with \( n \) units comes in two flavours, a bipolar network and continuously valued network. Let \( a_i \) be the state or the output of the \( i \)th unit, then for bipolar networks \( a_i \) is either +1 or -1, whereas it can assume any value between 0 and 1 for continuous networks. The network dynamics for the bipolar Hopfield network are:

\[
y_i = \text{Sgn}(\Sigma_j w_{ij} y_j - \theta) \\
= \text{Sgn}(\text{Net}_i)
\]  

(2.14)

The dynamic update of the states of the network is performed in two ways: synchronously and asynchronously. In the synchronous updating scheme all the units are updated simultaneously at each step in time. Asynchronous updating involves randomly selecting and adjusting one unit at a time. Each state of the Hopfield network has an associated energy value. This ‘energy’ does not relate to the real energy of any physical system, but to an objective function that the network is designed to minimise. Thus the network evolves to a state that lowers its energy value in terms of the network dynamics given by the previous equation. The energy function of the bipolar Hopfield network in a given state \( y = (y_1, y_2, \ldots, y_n) \) is defined by:
The central property of the energy function is that as the network state evolves according to the network dynamics, the network energy decreases until it eventually reaches a local minimum point. This local minimum functions as a basin of attraction where the network stays with a constant energy. When a set of patterns is stored in such a network, it functions as an associative memory. The network stores patterns by associating each pattern with a specific basin of attraction. However, there are severe constraints on the number of patterns that can be stored by such a network, and a Hopfield network can only cope with about \( 0.15n \) random patterns (see e.g. Haykin, 1999). Hopfield networks always evolve in a direction that lowers the network’s energy. This allows the network to deal with combinatorial optimisation problems because these can often be reformulated as an energy minimising function. One of Hopfield’s much celebrated achievements is to have designed a network that succeeds in solving a restricted version of the classic Travelling Salesman problem. In this problem a hypothetical salesman must visit a number of different cities and tries to compute the shortest route that he should travel in order to visit all the cities at least once. The problem is computationally complex because there is a combinatorial explosion of possible routes. For instance if there are 40 cities then the salesman has to cope with 40! different routes (see Section 5.5.1.1). The problem actually falls within the class of computationally intractable, or NP-complete problems, which do not have analytical solutions in some worst case scenarios (see Garey & Johnson, 1979). In Hopfield’s approach the connection weights of the network are determined by distances between cities on the salesman tour and the optimum solution to the problem is a fixed point of the dynamical equations (Hopfield & Tank, 1985).

Hopfield’s initial way of using the energy function led to a rather inefficient type of content-addressable network now called a Hopfield net. To store a vector, the weights are changed to reduce the energy of that vector so that stored vectors correspond to local minima in an energy landscape. To retrieve a vector from a noisy version of it, the network is put into an initial state and allowed to settle into an energy minimum, but only a few vectors can be stored without resulting in the creation of spurious local minima. Hinton and Sejnowski (1986) suggested a different way of making use of the energy function. Instead of settling to the nearest local minimum, the network should settle to a global (or near global) minimum subject to fixed boundary conditions representing the current processing task. The main problem with their suggestion is that the network can get stuck in local minima. Fortunately, Kirkpatrick, Gelatt, and Vecchi (1983) had shown that simple gradient-descent searches could often be improved by adding thermal noise that allows the system to make occasional uphill moves in the energy function. The best results are generally obtained by steadily decreasing the amount of noise as the system settles. This technique is called “simulated annealing”, and Geman and Geman (1984) showed that it can be
effectively used in image processing tasks. The ideas of Hopfield and the simulated annealing technique were combined in a network called a *Boltzmann machine*. In this network each unit has a stochastic decision rule governed by an ‘energy gap’ that depends on the states of neighbouring units and the weight coefficient on its connections. The unit turns on with a probability that is a Boltzmann distribution of this energy gap.

The development of the Boltzmann machine is an important milestone in ANN theory. It resulted in the construction of an important learning system that can be analysed theoretically, and which exploits a slow but effective learning mechanism. More significantly, it placed the connectionist research paradigm at least partly within a secure theoretical context. For example, the parallel between the energy model used by Hopfield and the Ising model (associated with spin glasses) helped to spread its popularity among the physics community and established the scientific legitimacy of ANN modelling (Anderson, 1995).

### 2.4.2 Kohonen’s self-organising nets

Apart from Hebbian learning, all the approaches discussed so far are examples of algorithms based on supervised learning. A given network is presented with an input pattern and an expected output pattern and the learning algorithms then typically makes changes to the set of weighted connections and internal activation values of the units in order to approximate the expected output pattern. However, there are also connectionist systems that are capable of learning patterns in data without the aid of a ‘teacher’, such networks are called unsupervised or self-organising networks. Biologists and psychologists are interested in self-organising networks because some researchers consider these networks to be biologically more plausible than the supervised category. The ideas for self-organisation were proposed around 1973 by von der Malsburg, and were followed by computer models for self-organisation a few years later (e.g. Willshaw & von der Malsburg, 1974).

Various rather complicated types of self-organising systems have since been developed. An elegant and simple architecture has been investigated by Kohonen at the University of Helsinki from the mid 1970s, and the resultant networks are now often called *Kohonen nets*. Kohonen’s work is inspired by the brain’s ability to construct spatial mappings of complex data structures, thereby capturing the topological relationships in the data. Much of the cerebral cortex is arranged as a two-dimensional plane of interconnected neurons, but it is able to deal with three-dimensional patterns. Kohonen tried to simulate this ability to exploit lower-dimensional representations by developing a data compression technique
called ‘vector quantisation’. The idea of vector quantisation is to represent multi-dimensional data in a much lower dimensional space, analogous to the operation of correspondence analysis or k-means clustering. In contrast to the layered perceptron, the units in a Kohonen net are arranged on a flat grid. All inputs are connected to every unit in the net, and feedback is restricted to lateral interconnections to immediate neighbouring units.

The learning algorithm organises the units in the grid into local neighbours that act as feature classifiers for the input data. The net is autonomously organised by a cyclic process of comparing input patterns to vectors stored at each node in the network. No target values are specified for the input patterns. Instead the network attempts to find the closest matching unit to an input, and increases the similarity of this unit, and those in the neighbouring proximity of the input. In the learning procedure the first step is to calculate the distance $d_j$ between the input and each output unit $j$ at a given time $t$, which can be obtained by:

$$d_j = \sum_{i=0}^{s-1} (y_i - w_{ij})^2$$

(2.16)

The learning algorithm selects the minimum distances, and then updates the weights for unit $j^*$ (the output unit associated with $d_j$) and its neighbours, defined by a “neighbourhood size”, $N_{r}(t)$, according to the rule:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)(y(t) - w_{ij}(t))$$

(2.17)

for $j$ in $N_{r}(t)$. The term $\eta(t)$ is a gain term that decreases in time thus slowing down the weight changes. This learning rule is very simple, and implements a form of competitive learning in which the idea is to move the weight vector so that it more nearly aligns with the input vector. The winning unit is the one whose weight vector is closest to the input vector. The result of training is to nudge the weight vectors of the winning units closer to the input pattern and in this way to localise the area of maximum activity. As can be seen, this learning rule is closely related to Hebb’s rule.

### 2.5 Training Multilayered Perceptrons: The Backpropagation Algorithm

Minsky and Papert (1969, 232) conjectured that networks that were composed of multiple layers of units
would be subject to many of the same limitations associated with the perceptron, and that the extension of research to more complex network structures would therefore be "sterile". In the 1980s this conjecture was shown to be false.

Although there has been constant research activity in the neural network field, much of this work was restricted to a few key researchers in psychology, physics, and biology, but the field suddenly exploded after the publication of the backpropagation learning algorithm for multilayer perceptrons by Rumelhart, Hinton and Williams (1986). The essentials of the backpropagation method were developed by Werbos (1974) as a mechanism for optimising the predictive ability of mathematical models. It was later rediscovered by Le Cun (1987), Parker (1985) and Rumelhart et al (1986). However, the current popularity of the algorithm is mainly due to the work by Rumelhart and his colleagues. The algorithm was initially defined for a feedforward network with three or more layers and an arbitrary number of input and output units, but can also be extended to more complicated architectures such as networks with recurrent connections. Some types of recurrent networks have proven to be useful in coping with the temporal information that underlies processing and interpreting aspects of language (see section 5 in this respect). In the typical, vanilla backpropagation case, a network is presented with a set of input patterns as well as a corresponding set of target patterns, and is required to learn the mapping from input to output. The algorithm therefore entails a form of supervised learning, like the perceptron, but unlike the Kohonen nets which use an unsupervised learning procedure.

The network computes a weight adjustment rule which changes weights in the direction of the steepest decline on the error surface as defined by the total error (summed across patterns). Rumelhart and his colleagues showed that the following rule can be used to change the weights for each pattern (see Appendix B for a brief sketch of the derivation of the learning rule, called the "generalised delta rule").

\[ \Delta w_{ij} = \eta \delta y_i \]  
(2.18)

where for units in the output layer, \( \delta \) is defined as:

\[ \delta_j = (t_j - y_j) f'_j(Net_j) \]  
(2.19)

and for units in the hidden layer it becomes:

\[ \delta_j = f'_j(Net_j) \sum_k \delta_i w_{kj} \]  
(2.20)
As shown in equation (2.20), the deltas of the hidden units are computed in terms of the deltas at an upper layer, delta \( k \). The derivation of the learning rule is described in Rumelhart, Hinton & Williams (1986), and a short sketch of the derivation, but one which is included just for completeness and lacks the ‘meat’ of the original, is presented in Appendix B. This learning procedure bears an obvious resemblance to the delta rule employed by Widrow and Hoff, but the main difference between these two methods lies in the way in which the deltas are calculated. While the deltas in the Widrow-Hoff algorithm are obtained by simply calculating the difference between the target value and network output, the delta in the backpropagation algorithm is a function of the difference between the target and output values and the first derivative of the unit. The latter procedure can be applied to any multilayered network, and is therefore often designated as the generalised delta rule.

The backpropagation learning algorithm involves the following steps:

1. All the weights in the network are randomly initialised.
2. The input and target patterns are entered which entails entering \((I_1, I_2, ..., I_i)\) and \((t_1, t_2, ..., t_k)\).
3. The network generates output patterns based on the current input and weight settings.
4. The generated output patterns are compared to the target patterns and the target value associated with each output unit is used for calculating the delta for each output node.
5. After calculating the deltas in the output layer, the deltas for the hidden layer are computed with the aid of equation (2.20).
6. All the weights are adjusted by using equation (2.18).
7. Step 2 is repeated until all the patterns have been entered. Once all the patterns in the data set have been presented to the network, one epoch of training is said to have been completed.

Because the error function has a non-linear form with respect to the parameters (the weights in the model), the backpropagation procedure uses gradient descent to minimise the error function. The algorithm entails two passes, a forward pass and a backward pass. During the forward pass the network computes the output value associated with each output unit based on the input pattern and the weight matrix in the network. The network then computes the deltas starting with the units in the output layer, and propagates the deltas backward to the hidden layers. This latter phase is called the backpropagation phase and it is largely due to this ‘backpropagation’ process that the algorithm owes its name.

Unlike the perceptron, the backpropagation learning procedure is not restricted to linear functions, but can also be applied to nonlinear functions. For instance, the network in Figure 2.4 below can compute the XOR function, and therefore succeeds in solving Minsky and Papert's (1969) celebrated XOR
Figure 2.4. A solution to the XOR problem

The fact that the network in the figure above solves the XOR problem can be verified by inspection, but two input cases are worked through below for expository purposes. We assume that the network transmits 1 when the activations arriving at it exceeds its threshold value, and 0 otherwise. When presented with the pattern (0,1) the hidden neuron will receive the input activations \((1 \times 0 + 1 \times 1 = 1)\). Since this value does not exceed its threshold of 1.5 it will not transmit any activation to the output unit. The actual activations received by the output unit are therefore only \((1 \times 0)\) from the left unit, and \((1 \times 1)\) from the right unit. Hence because these summed activations exceeds its threshold value of 0.5, it will transmit an activation of 1. In contrast when the net receives the pattern (1,1) the activations impinging on the hidden unit will exceed its threshold and cause it to transmit an activation of -2.0 to the output unit. The summed total arriving at the output unit (-2 from the hidden unit + 1 from the left unit + 1 from the right unit) will not exceed its threshold so that the unit will transmit the value 0. Notice that the multilayered network is capable of solving this problem because it incorporates two novelties (compared to the perceptron). A nonlinear output activation function which is required for the backpropagation training regime, and the inclusion of one or more hidden layers, which enable the network to develop an internal representation or memory for the patterns in the input data.

Learning is sometimes very slow in backpropagation nets and there is also no guarantee that the algorithm
will succeed in locating the global minimum (i.e. the lowest level in the error landscape), because the learning process can sometimes get stuck in a local minimum (i.e. a crevice at some intermediate level in the landscape). One popular approach to solving the latter problem is to employ a momentum term (denoted by $a$ in the equation below) as suggested in Rumelhart et al. (1986). The procedure entails replacing equation (2.18) by equation (2.21) below:

$$
\Delta_{t+1}w_y = \eta \delta_y y_j + a \Delta_tw_y
$$

(2.21)

In equation (2.21) the change in the weights at time $(t + 1)$ is computed as in equation (2.18), but by also taking into account the previous change at $t$. This has the consequence of moving weights in the same direction as that of previous changes so that the network is more likely to skip over small local minima.

Apart from the occasional slowness of the convergence process, the backpropagation approach is plagued by a few other problems. For instance, McCloskey and Cohen (1989) found that a network trained by backpropagation on a specific function and then exposed to a new function quickly forgets everything that it has previously learned. In order to learn the new function, the network must first unlearn what it has already learned. Moreover, there has been criticism levelled at the biological claims made about backpropagation, because the learning procedure is biologically unrealistic - real neurons do not propagate error signals backward. Because of these difficulties various modifications and improvements to the basic algorithm have been proposed. Most of these modifications are rather technical in nature (see Hertz, Kroch & Palmer, 1991 for details) and will not concern us here, but we shall consider some of these alternative proposals in later chapters.

Despite these difficulties, the algorithm yields a powerful learning mechanism. In fact, there is proof that a neural net with two hidden layers, trained with the backpropagation procedure, is capable of approximating to an arbitrary degree of accuracy any continuous, deterministic function provided there are enough exemplars to train the network on and that the network has a sufficient number of hidden units (Hornik, Stinchcombe & White, 1989). A function here is taken to be a mapping from the real-valued domain $\mathbb{R}^n$ to the set of real numbers $\mathbb{R}$, and the notion of "approximating to an arbitrary degree of accuracy" means that although the network will not generate the exact function values it will keep on getting closer and closer to these values until some predetermined stopping criterion is reached.

The design of such a network is subject to some practical constraints, requiring a careful choice of parameters such as the topology, number of units, learning rate and, of course, the availability of adequate training data. Furthermore, there are some practical limitations on the learning capacity of a
typical three-layered (i.e. a network with one hidden layer) network (see Masters, 1993, p. 86-87):

- If the function comprises a finite collection of points, the network will be capable of learning it;
- If the function is continuous and defined over a compact domain the network will be able to learn it. A compact domain implies that the inputs have definite bounds;
- Discontinuities can be tolerated under all conditions likely to occur in practice (as opposed to in theory), and functions that are not defined over a compact domain such as when the inputs are normally distributed random variables can also learned.

If a feedforward neural network proves incapable of learning a given function, the most likely reason is that the function fails to obey the condition of a compact domain. As Masters (1993, p. 87) observes, it is unreasonable to expect an ordinary feedforward network (recurrent networks have some special properties that we shall consider later) to approximate a function whose inputs range to infinity. Nevertheless, even such functions can be learned by implementing a trigonometric output activation function which causes the network to behave like a Fourier approximator. Thus for all practical purposes, a neural network can be designed that will in principle prove capable of learning a given input-output mapping, and any practical difficulties that such a network might encounter in learning the function is due to insufficient training, or to a design that lacks enough hidden units, or to the fact that the function is not deterministic, rather than to computational limitations inherent in the model itself (i.e. the backpropagation learning mechanism).

Given these powerful learning and generalisation capabilities of multilayered networks trained with backpropagation algorithm, it is clear that a mature era in connectionist theory has been reached. These neurocomputational systems build on traditional associative mechanisms, but they transcend the older associative approaches because they contain nonlinear processing mechanisms. Notice for instance that in this modern conception of associative computation, association is not a function of a connection between symbolic representations in memory, but of the similarity between stored patterns of activity in a complex network of interrelationships. This computation of similarity is sometimes held to involve some sort of prototype extraction in terms of which input patterns representing exemplars of a category are compared to stored prototypes of the relevant category in memory (Churchland, 1995).

Still, the associative tradition had some impact on the development on the theory of neural computation because a connectionist model can be regarded as a specific implementation of an associative computational system. However, while associationism is evidently a significant historical force in the development of connectionism, the main impetus for the current surge of interest in neural network
models derives from attempts to model cognitive and brain functioning. Hence, in the next section we shall leave our historical survey and move on to consider the relevance of the connectionist approach to cognitive theory.

\section{The Cognitive Science Context}

The major turning point in the neural net research paradigm occurred with the formulation of training procedures for multilayered networks. With the discovery of the backpropagation algorithm, connectionist models acquired significant computational power which revived the interest of cognitive researchers in these systems.

\subsection{The PDP era: Some 'early' network models}

It is perhaps significant that the first broad dissemination of the backpropagation algorithm was via the PDP books in which its application to cognitive modelling was demonstrated. In one of these applications, which has since become famous or perhaps 'infamous', Rumelhart and McClelland (1986) developed a network to simulate the acquisition of the past tense form of verbs. It is well documented in the psycholinguistic literature that young children go through a phase during which they use the past tense form of irregular verbs correctly, and then pass through a second stage when they apply an incorrect past tense to these verbs. They only use the past tense forms consistently correct in third stage. The learning follows a U-type curve, starting correctly, dropping to error-prone utterances, and then improving again. The standard explanation offered by psycholinguistics for the decline in performance after initial correct usage of past tense forms, is that children first repeat the past tense forms by rote, and then, in a second stage, begin to learn the rules underlying past tense formation. During the second stage they generalise the past tense form of regular verbs incorrectly to irregular verbs. Thus, on this account, a child might say \textit{goed} (go + ed) by analogy with the regular past tense form of verbs such as \textit{play} (i.e \textit{play + ed}). The third stage (of correct usage) occurs when children have learned the exceptions to the rules.

During learning, McClelland and Rumelhart's network reproduced the typical U-curve that children exhibit in their learning of the past tense form. Furthermore, the network was capable of generalisation and could compute a past tense form for verbs that were not part of the training set (i.e. it could deal with novel input). Although the network was criticised for employing a somewhat idiosyncratic representation of the input (Wickelfeatures) which some researchers claim may have biased the network during its
learning, the study is generally regarded as significant because language researchers have always assumed that learning past tenses is a quintessentially rule-based activity, yet the network learned this task without incorporating any rules. The PDP books also contained a number of other applications of neural network techniques to model cognitive abilities such as memory, schema abstraction and language processes (we shall return to consider some other applications in Chapter 4). Furthermore, at about the same time, Sejnowski and Rosenberg (1986) constructed a neural net called NETtalk that learned to produce speech output from written text. They trained the network to learn grapheme-phoneme associations on a set of 1000 words, and achieved a 95% success criterion after 50 000 trials. The model also generalised relatively well, giving about 77% best guesses when applied to a large dictionary of 20 000 words. Sejnowski and Rosenberg (1986) did a cluster analysis on the hidden unit activations and found that the network appeared to distinguish between consonants and vowels, possibly because in English spelling sound correlations across different words are more consistent for consonants than for vowels.

However the most remarkable aspect of the model was that it actually went through what sounded like a babbling phase, which is a characteristic phase in the early language acquisition in children (see Taylor & Taylor, 1990, pp. 242-243). This human-like aspect of its learning, plus the relative ease with which the network learned the grapheme-phoneme mapping, did much to promote ANN modelling among researchers in a variety of disciplines. Prior to NETtalk orthographic-to-speech conversion involved complex algorithms, and the fact that a program could learn 'by itself' to do the conversion, and outperform the prevailing serial algorithms in the ease with which it acquired this mapping, caught the attention of researchers. Following the success of these models, a growing number of researchers began to view multilayered networks (typically trained on the backpropagation procedure), as a framework for developing theories of human cognitive processes. This connectionist framework is probably best cast not as a list of specific detailed assumptions, but as a set of basic principles and some general guidelines that provide some constraints on the type of models that can be called 'connectionist'. More specifically, this framework is based on a conception of cognitive architecture comprising:

☐ A neural-like level of mental representation which lies below the symbolic level. In the next chapter, Smolensky's (1988) work will be discussed. He has suggested that the connectionist level should be called a "subsymbolic level", because in the typical ANN system of representations, a conceptual representation is not a specific, local object or an all-or-none entity which is either active or not. Instead the representations are manifested as general patterns of activity that spread over all the active units in the network.

☐ In connectionist systems information processing involves a large network of simple neuron-like
processing units operating in concert. This parallel processing architecture bears some resemblance to the functioning of the brain and to the relaxation techniques of statistical mechanics, and differs from the serial, rule-based algorithmic procedures inherent in symbolic, classical-level cognitive approaches.

The incorporation of a learning function which typically involves a form of gradient descent on a multidimensional error landscape. As described in the previous section, most connectionist systems learn by computing small changes to the connection weights in a network and attempt to minimise the output error in order to approximate the relevant input-output mapping. Furthermore such systems are typically capable of generalisation because after exposure to training patterns belonging to a given category, they will be able to generalise to an unfamiliar but similar pattern belonging to the same category. In general, classical theorists have concentrated mostly on delineating aspects of the processing mechanism and much less on issues of learning. Moreover, the notion of learning implies adding new representations to a classical system which presents a number of philosophical and theoretical difficulties (which will be discussed in the next chapter).

This in broad detail is the connectionist approach in cognitive science. This approach highlights the role of parallel processing mechanisms and differs in several respects from the prevailing classical or symbolic paradigm which will be considered in the next chapter. However, simply offering a new formalism is not sufficient grounds for endorsing connectionism, we need to show that it actually advances our understanding of aspects of human cognition by yielding new insights (i.e. insights not forthcoming from the symbolic paradigm). We now turn to consider the specific contribution of connectionism to cognitive theory in more detail.

The discovery of the backpropagation learning procedure had a significant impact on cognitive researchers who quickly began to accept ANNs models as a welcome adjunct to their repertoire of technological tools, and the use of ANN models is now firmly entrenched in psychology, cognitive science, artificial intelligence and various other academic disciplines. However, despite the widespread use of, and general appeal that connectionist models have for many researchers in the cognitive sciences, their 'proper' place in psychology and the other cognitive disciplines is still not clear (see Smolensky, 1988). There are various reasons why the status of connectionist modelling in the cognitive sciences remains problematic:

There is a general difficulty associated with interpreting theories of cognition that is probably
endemic to the cognitive sciences. The so called ‘cognitive sciences’ (which includes cognitive psychology, computer science, philosophy, linguistics, and neuroscience) are based on rather disparate assumptions both about the goals of research into human and machine intelligence, and about the appropriate methodology which should be applied in pursuit of these aims. There are also conflicting opinions as to whether it is worthwhile pursuing such goals at all, and whether they can be coherently formulated. The confusion inherent in the field renders truly objective assessments of new theoretical contributions somewhat problematic, leading to a number of very different and often conflicting interpretations of the theoretical importance of neural network research in the cognitive domain.

Related to the above point is the fact that the use of ANNs for cognitive modelling has become associated with a number of rather murky philosophical issues. The prevailing approach in the cognitive sciences (which will be set out in detail in the next chapters) is based on the assumption that the human cognitive system can be characterised in information processing terms, and a computer metaphor is used to drive cognitive theorising. However, it is not clear whether ANNs should be subsumed under the information processing umbrella, and whether their mode of processing can be interpreted in traditional computational terms based on standard theoretical models deriving from mathematical logic, such as a Turing Machine. One of the main issues at stake is whether ANNs should not rather be viewed as instantiating a new set of models which diverge from the orthodox information processing models in crucial respects (and indeed to clarify what these ‘crucial respects’ are).

Large multilayered networks are technically very complex so that analysing the cognitive aspects associated with such nets is far from straightforward. Hence even if a connectionist simulation of a given cognitive process can be developed, the way in which such a network enhances our understanding of the attendant cognitive processes may not always be apparent to researchers. Some researchers have extolled the virtues of ANNs in simulating cognitive processes, without addressing the issue of whether these simulations help us to gain a better theoretical understanding of the relevant phenomena.

As already mentioned, ANNs are practical tools which can be used in a variety of everyday applications in engineering and finance, such as speech processing, pattern recognition, time series analysis, and data mining. In these contexts, neural nets provide an alternative to existing statistical techniques such as multivariate non-linear regression, correspondence analysis, and principal component analysis. At the same time, neural nets yield a framework for simulating
cognitive phenomena, and in this respect they go well beyond standard statistical methods and have much closer affinity to artificial intelligence technologies. This distinction between neural nets as a technical tool that can be used for testing theories about aspects of cognitive phenomena, and as an actual model of the relevant phenomenon, is not always appreciated in the literature.

Moreover, it is important to realise that ANN modelling is an evolving paradigm, new network models are proposed at a constant rate, and some older criticisms of neural network models may now attack a strawman. The set of connectionist techniques, simulation toolboxes and models driving these technologies constitute in many respects a shifting horizon. It is therefore important to be clear about whether a criticism of the connectionist paradigm as such is not actually only directed at a particular ANN model without necessarily encompassing the whole enterprise. This makes it very difficult to assess the real status of connectionist modelling, so that there is a decided lack of clarity about the foundations and theoretical relevance of the approach in cognitive science. In the next chapter we begin to look at these foundational issues.
3

CONNECTIONISM, COMPUTATION, AND COGNITION: FOUNDATIONS

The main theme of this thesis is to work towards a clearer understanding of connectionism as a theoretical option in cognitive science by isolating the salient properties of this approach, by describing in more detail what exactly a connectionist cognitive architecture entails, and by addressing the issue whether connectionist modelling has helped to enhance our understanding of human cognition. However, before considering connectionism in more detail, it is necessary to sketch the general theoretical context, the so-called ‘Computational Theory of Mind’, which used to dominate the cognitive sciences and in which connectionism evolved. Most of this chapter is devoted to the latter task, after which some of the basic concepts and assumptions of the connectionist conception of cognitive systems are introduced.

3.1 COGNITIVE THEORY AND INFORMATION PROCESSING

As noted above, the aim of this thesis is to consider the contribution of connectionism to cognitive theory. Unfortunately, the psychological and cognitive sciences lag behind mature sciences such as physics in theoretical sophistication. There are few well-developed formal theories in this area (e.g. Newell, 1990 is an exception), which tends to be characterised mostly by a collection of fairly ‘local’, often very intuitive, theories addressing various aspects of cognition such as memory, attention, vision, or language. In fact, the field is marked by numerous disputes about what exactly the domain of inquiry is, what methods should be used to develop and test theories of psychological processes, and what constitutes an acceptable psychological explanation (Dawson, 1998, p. 1). There are also differences in opinion about the nature of formalisms and even whether psychology lends itself to formal inquiry; thus, Lakoff (1987) rejects the positivist assumptions underlying some formal approaches outright. To some extent the psychological sciences are still very fractured, and inchoate. Nevertheless, in mainstream cognitive psychology and cognitive science, there is at least some consensus about the way in which the development of theories of cognition should be pursued, about underlying assumptions, and about the principles and explanatory objectives governing such theories (Dawson, 1998, p. 3). It is therefore useful to begin by surveying this general approach.
The received doctrine in the cognitive sciences (which groups together, inter alia, cognitive psychology, philosophy of mind, artificial intelligence, neuroscience, and linguistics) is known as the “information processing paradigm” (Lachman, Lachman & Butterfield, 1979) or “computational theory of mind” (Horst, 1995). The approach draws from a computational metaphor, and likens the operation of the mind to a computational system which performs information processing tasks on cognitively represented knowledge structures. According to this view, cognitive research is mainly concerned with the internal processing mechanisms and knowledge structures characterising human psychological processes, and much less with external aspects such as social and environmental aspects affecting behaviour. The emphasis is on discovering “how the mind works”, and research is directed at describing the cognitive structures and mechanisms governing human, animal, and machine intelligence (Lachman, Lachman, & Butterfield, 1979; Pinker, 1997). The advantage of the information processing perspective is that it provides a way of conceptualising higher mental functions such as perception, memory, learning, reasoning and language understanding. For instance, according to this view a perceptual process such as object recognition involves a mapping between incoming visual information and structural descriptions stored in memory, with the implication that object identification occurs when a suitable match is computed between the visual stimuli and the knowledge stored in memory. The main assumption is that internal information processing mechanisms mediate everyday psychological tasks such as perception, reasoning, and language interpretation.

To some extent, the information processing paradigm is a fairly general framework, functioning much like an umbrella theory, under which more specific interpretations of cognitive information processing are subsumed. These interpretations articulate specific assumptions about the nature of processing mechanisms, about the format of cognitive representations, about the nature of the formalism used to develop models of aspects of cognition, and about the associated cognitive architecture. Unfortunately, the concept of information processing itself is often used in a somewhat vague and metaphorical manner, without a clear indication of either its underlying assumptions or specific theoretical postulates. Hence, it is useful to begin by trying to develop a clearer picture of what the information processing perspective entails.

3.1.1 Roots of the information processing perspective

The basic premise of the information processing approach in psychology is that cognition is a form of computation. But what is meant by “a form of” and what is a “computation”? Intuitively a computation is a systematic mapping between interpretable inputs and outputs. Such mappings can be either one-to-one
or many-to-one between elements of one set, the range, and another set, the domain. A computable mapping (or function) is one that can be specified in terms of some rule or other and can be roughly characterised in terms of what must be done to the first element to get the second. For instance, given input such as "3 + 5" a computation produces the output 8, and when the input is "7 + 3" the same computational process generates 10 as output. Here the input and the output involve quantities and the mapping between them is systematic so that it can be described by a function, namely addition. A computation need not involve numbers, but can also be defined over character strings so that determining the length of a string of characters, or counting the words in a sentence, or determining the grammaticality of a sentence are all computations, because in each case it is possible to generate a unique answer as output.

A computation can therefore be conceptualised as a systematic or "effective" procedure for mapping appropriate inputs to outputs. Any mapping defined over interpretable inputs constitutes a computation, the sole conditions being that the computational mechanism should have the requisite power, and access to a relevant set of instructions to generate the output.

The description above helps somewhat, but is admittedly rather informal. Thus the concept of an effective procedure requires further elaboration. We also need to know which set of mappings can be construed as computations, and how human cognition fits into this computational scheme. A significant achievement of mathematical logic, and cognitive science (see Dawson, 1998; Wells, 1999), is to have arrived at a more rigorous and insightful formulation of the concept of a computation, and of the attendant notion of an effective procedure, which makes it possible to address these issues. To appreciate this contribution to both computational and cognitive theory, a slight digression is necessary in order to introduce the technical notion of a Turing Machine.

### 3.1.2 The Turing machine model of computation

Turing presented the idea of a Turing Machine (TM) in his paper "On computable numbers" (1936) as a solution to the mathematical problem of furnishing a general characterisation of the class of functions that admit of computable (i.e. algorithmic) answers. In 1900 Hilbert had listed a number of unsolvable mathematical problems which he regarded as worthy of mathematical research. Turing became interested in the 23\textsuperscript{rd} Hilbert problem which involves the construction of a mechanical procedure for testing whether an arbitrary chosen formula in first-order predicate logic is a theorem. An example of such a formula would be: For every \(x\) there is a \(y\), such that \(y\) is \(R\) to \(x\). When \(R\) is interpreted as "larger than", this statement articulates a theorem about the natural numbers. A mechanical test for determining whether the statement is in fact a theorem is a decision procedure, and Turing's concern was to establish whether such
a decision procedure could be formalised. He conceived of an abstract machine, called a Turing Machine (TM), which would be capable of performing any mathematical task that one would intuitively think of as being susceptible to such a decision procedure. He was then able to recast Hilbert’s 23rd problem in a mathematically precise form as that of whether one could design a Turing Machine which, given a specific formula of predicate logic, could determine whether it constitutes a theorem.

It is important to realise that a TM is a mathematical abstraction rather than a piece of hardware. Nevertheless, it can be envisioned as a physical machine with a control device containing a read-write head (rather like a typewriter), that acts on a tape marked off in squares, each of which is either filled with a symbol or a special ‘blank’ symbol. The machine reads off information from the tape one square at a time, and also writes to the tape. The act of writing symbols enables the machine to erase previously written symbols on the tape (e.g. by writing the blank symbol over them). The tape itself can be imagined as extending infinitely to the left and the right, with the TM’s control head moving left and right on the tape, as illustrated in Figure 3.1 below:

![Control Head](image)

**Figure 3.1** The control head of a TM scanning a portion of the input tape. For the purposes of this illustration the tape must be imagined to extend infinitely towards the right.

Formally, a TM can be viewed as a quadruple \((S, \Gamma, M, \delta)\) where \(S = \{s_i \mid 1 \leq i \leq m\}\) is a set of internal states, excluding the halt state \(h\), \(\Gamma = \{a_i \mid 1 \leq i \leq n\}\) is an alphabet of symbols including two special symbols, a blank symbol which represents an empty cell entry (e.g. \(B\)), and a symbol “\(h\)” which instructs the machine to halt. In the definition above, \(M = \{L, R, N\}\) is a set of movement indicators indicating whether the machine should move left, or right or execute no movement at all, and \(\delta\) is a (partial) function from \(S \times \Gamma\) to \(S \times \Gamma \cup \{L, R, N\}\). The set of internal states has a starting state, \(s_1 \in S\), and the set of arguments and values of \(\delta\) specify a machine table for a given TM. This table contains instructions of the form:

\[\delta(s_i, a_k) \rightarrow (s_n, a_k, m_i)\]

which is to be read that when the machine is in state \(s_i\), scanning symbol \(a_k\), it must rewrite \(a_k\) as \(a_k\), enter state \(s_n\), and move in direction \(m_i\). When \(s_i = h\), or if the value of \(\delta(s_i, a_k)\) is undefined, the machine will halt (see Partee, ter Meulen, & Wall, 1990, p. 507-514).
The actions executed by a TM involve fairly basic operations such as “move right one cell”, or “move left one cell”, or “print symbol on tape”, or “erase old symbol on tape”, or “change content of internal state. Thus, Table 3.1 below, taken from Dawson (1998, p. 30), displays a simple set of machine instructions that will enable a TM to perform addition. In the table the symbol “B” denotes a blank, and the instructions pertain to two strings of numbers in unary notation, with a blank separating them, which the machine must add together in order to generate their sum as output.

<table>
<thead>
<tr>
<th>Current state of the machine</th>
<th>Symbol read from tape</th>
<th>B</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Move 1 cell right</td>
<td>Move 1 cell right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adopt State 1</td>
<td>Adopt State 2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Write 1 to tape</td>
<td>Move 1 cell right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Move 1 cell right</td>
<td>Adopt State 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adopt State 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Move 1 cell left</td>
<td>Move 1 cell right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adopt State 4</td>
<td>Adopt State 3</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Stop</td>
<td>Write B to tape</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Don’t move</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Adopt State 4</td>
</tr>
</tbody>
</table>

**Table 3.1** Example of a set of actions carried out by a Turing Machine

The table above is merely intended to illustrate the typical actions that a TM performs, and as can be gleaned from the cell entries describing the intended actions, it is conceptually a rather simple device. It is therefore remarkable, and also somewhat counter-intuitive, that a TM turns out to be computationally extremely powerful in terms of the range of functions (i.e. input-output mappings) that it can compute. For instance it has been shown that this simple machine can execute mappings from integers to integers and that it can handle all the common arithmetic operations (see Penrose, 1989, p. 35-70 for some examples). In addition, a TM can compute logical functions, and can determine whether a string of symbols constitutes a sentence of a given (formal) language. It can recognise sentences of the set of recursively enumerable languages, such as the context-sensitive languages (which is thought to include the set of human languages) in this way (Partee, ter Meulen, & Wall, 1990, p. 511-515).
Intuitively, the operation of the machine corresponds to three basic functions; the successor function (accessing the next cell on the tape), the set of projection or identity functions, (recognising a given symbol on the tape), and the zero function (erasing a symbol, leaving a blank in its place). Together these three functions constitute the class of partial recursive functions, and they are significant in that more complex functions can be constructed from them in a relatively straightforward manner by means of processes such as composition, recursion and minimisation (Hamilton, 1978, p. 133-135). Moreover, they are easily translatable into programming constructs in that the operations of composition, recursion, and minimisation correspond to subroutines, simple iterative loops, and while-loops respectively. It is a basic tenet of structured programming that any computer program can be constructed from these three constructs and a set of basic functions and data types (see e.g. Johnson-Laird, 1983, p.22). One of the implications stemming from the power of the machine and from the description of its functioning in terms of the set of partial recursive functions, is that these functions may capture what we commonly understand under the notion of an effectively computable procedure (and as we shall see below, this is in fact stated in the Church-Turing thesis). Note that the qualifier “effectively” is added to signal that the machine will eventually halt, producing its output. For obvious reasons a function for which the machine will never stop, is not computable because the machine will never generate an output as response. There are interesting functions, such as the halting problem, which involve establishing whether a particular program will halt for a given input, that are not computable in the sense defined above (see Jones, 1997, p. 77).

Turing advanced the concept of computation further by formulating a special TM called a Universal Turing Machine (UTM), which is capable of emulating any other TM. The bridge from a specific TM to a UTM was achieved by including in the representational format of the machine tape, information concerning the data (i.e. symbols to process) as well as the specific TM that performs the processing, and by further incorporating in the machine head, a table setting out the instructions to be executed by the various TMs. Essentially a UTM operates by first reading off the tape the machine description (i.e. identity of the TM to emulate) and then consulting the machine table for the instructions to act out or simulate the relevant TM (Partee, ter Meulen, & Wall, 1990, p. 520-522). These instructions are then used as a basis for processing the symbols and associated instructions on the tape. Hence, because a UTM emulates other TMs by consulting an internal look-up table of machine instructions (i.e. a program), it can be regarded as a universal computing machine, and as a conceptual archetype of the contemporary programmable computer.

Contemporaneously with Turing’s development of the TM, other mathematicians also explored formalisms aimed at clarifying the notion of computability. Examples of such formalisms are the lambda calculus
(Church, 1936), production systems (Post, 1946), and recursive functions (Kleene, 1952). Significantly, these formalisms (and those developed afterwards), yielded essentially the same model of computation in that they isolate exactly the same set of functions (i.e. the set of partial recursive functions) as Turing's model. The fact that different models of computation converged on the same set of functions resulted in a well-known thesis called the "Church-Turing thesis", which states that the class of effectively computable functions coincides with the set of partial recursive functions. The thesis is usually expressed in terms of the two statements below (see e.g. Jones, 1997, p. 9):

- All reasonable formalisations of the intuitive notion of an effective procedure (or computability) are equivalent and isolate the same set of functions, the set of partial recursive functions. The thesis thus posits an equivalence between the set of partial recursive functions, and the set of computable functions.
- A TM is a reasonable formalisation of the concept of an effective procedure, and hence of computability.

Note the important implication of this thesis. Since it is strongly suspected (not proved, because the Church-Turing thesis is not a theorem) that the different models of computation are all equivalent, and thus of equal power, it appears that a TM is as powerful a computational device as there can be. It may not be as efficient or easy to program as a modern computer, but it is just as powerful in terms of the set of input-output mappings that it can compute. Indeed, a TM exhaustively defines what computation is all about.

3.1.3 From computation to cognition

The Church-Turing thesis presented in the previous section characterises computation in terms of the mappings associated with three rather basic functions (i.e. the set of partial recursive functions), which in turn describe the behaviour of a TM. A TM's computations are defined over symbols entered in the cell entries of the tape; it generates input-output mappings by manipulating symbols on the tape. As we shall see in the next section, Newell and Simon (1972) maintain that the whole class of intelligent systems is definable in terms of such symbol manipulation capabilities. Moreover, a TM's operation highlights some basic properties of (universal) computing devices:

- Its mode of functioning is extremely simple, involving very basic logical actions such as moving and erasing symbols on a tape;
- It computes by simply following the instructions, or program, encoded in its machine table, and
the whole computational process unfolds in a purely mechanical manner;

☐ The machine’s operations are purely syntactic in nature, based on a given symbol’s position on the tape, its formal properties (shape, etc), and involving (syntactic) actions such as moving a symbol or substituting it with another one

☐ The Church-Turing thesis links the notion of computability to that of the (partial) recursive functions, with the implication that any mechanism that lacks the ability to deal with recursion is also limited in its computational power. As we shall see below this is why cognitive scientists tend to be critical of purely associative devices;

☐ A TM in its universal incarnation, the UTM, has a relatively simple structure, but encompasses an incredibly large repertoire of behaviour. It is in principle capable of executing any computable function.

Taken together, these aspects illustrate how a small number of elementary information processing operations can be combined to generate very complex behaviour. It is this latter aspect, in particular, that makes the concept of computation attractive to cognitive scientists. It suggests that the mind might likewise be composed of a set of basic computational or symbol manipulation structures, and consequently that it may be characterised as an abstract computational machine or automaton, analogous to a TM. It is important to note though that the mind may be like a TM, and not that it is a TM. For instance, the human mind is obviously a more efficient processing mechanisms than a TM because it incorporates extra features such as a the ability to accept complex expressions, a random access memory, and the ability to generate output as a probabilistic function of the input (Chalmers, 1996). Nevertheless, the core intuition is that the human mind can be construed as symbol-processing device, and that it bears a resemblance (at least in some formal sense) to other computational machines such as a TM.

The notion that the mind can be conceptualised as such a machine is a premise that largely defines the classical conception of cognition. However, before taking a closer look at this classical approach, two further concepts need to be introduced, the first relates to computational power, and the second to an explanatory strategy called “functionalism”.

3.1.3.1 Language recognition and computational power

A TM is essentially a symbol processor, and its computational abilities extend to symbolic systems such as language; it can process linguistic as well as numerical information. Actually, this claim about the machine’s ability to cope with language should be taken with a grain of salt, because it is much too clumsy
to function as a language processor in any real-life application. However, in theory a TM has the ability to function as a language acceptor. What this means is that if the machine is presented with a string of characters, it can determine whether the characters belong to one of a range of possible formal grammars. Turing machines are very powerful as language acceptors and can cope with the recursive languages which lie near the top of the Chomsky hierarchy of formal languages (Partee, ter Meulen, & Wall, 1990, p. 562). Chomsky presented a formal analysis of the conditions under which languages could be learned and constructed a hierarchy of languages, as well as their learnability criteria. The criteria are set out in terms of automata (formal, computational mechanisms) that are capable of learning a particular languages. The languages are ordered in terms of complexity, ranging from the most complex (Type 0) to the least complex (Type 3). The hierarchy is shown in the table below (adapted from Cohen, 1986, p.740-741).

<table>
<thead>
<tr>
<th>Type</th>
<th>Name of language</th>
<th>Acceptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Phrase structure (recursively enumerable)</td>
<td>Turing Machine (with an infinite tape)</td>
</tr>
<tr>
<td>1</td>
<td>Context sensitive</td>
<td>Turing Machine with a bounded tape</td>
</tr>
<tr>
<td>2</td>
<td>Context free</td>
<td>Pushdown automaton</td>
</tr>
<tr>
<td>3</td>
<td>Regular</td>
<td>Finite state automaton</td>
</tr>
</tbody>
</table>

**Table 1.1**  
Chomsky’s hierarchical ordering of formal languages in terms of the complexity of the structures they can learn.

One of the reasons why a TM is such a powerful concept, and why it is deemed to be of interest to cognitive scientists, is that it can cope with natural languages, whereas some other machines, such as a Finite State Automaton (FSA), cannot. An FSA is similar to a TM, but does not have an infinite tape, and moves in a forward direction only, scanning one symbol at a time. Bever, Fodor, and Garret (1968) illustrated this limitation of an FSA in a proof based on a simple language comprising strings composed out of two letters “a” and “b” only, and generated by the grammar b"ab". Some valid sentences generated by this grammar are a, bab, bbbabbb, and so on, whereas strings such ba, baab, or ababa are invalid. They showed that an FSA cannot correctly generate strings belonging to this grammar. The actual proof that an FSA cannot cope with the language in question is based on a fairly straightforward application of the Pumping Theorem for finite automaton languages (see Partee, ter Meulen, & Wall, 1990, p. 471-473). Using the theorem it can be shown that an FSA cannot recognise palindromes (which requires the ability
to cycle back to check whether the required symmetry holds). An FSA cannot deal with recursion, and therefore lacks the necessary computational power to correctly classify strings as valid or invalid in terms of the grammar. Bever et al. (1968) then point out that the above simple formal language directly reflects aspects of natural language such as long range dependencies and an embedded clause structure, as illustrated by the two examples below:

☐ Long range dependencies are grammatical relations between words that are divorced from one another in a sentence. For instance in the sentence, John was angry with himself because he forgot his wallet at home, the possessive pronoun his refers back to the noun phrase John in the main clause. The formal language exemplifies such dependencies, because to establish whether the required symmetry between the two strings of b's holds, the first b must be related to the last b in the sentence, as shown below:

\[
\begin{align*}
  b & \rightarrow John \ was \ angry \ with \ himself \ because \ he \ forgot \ his \ wallet \ at \ home \\
  \downarrow & \hspace{1cm} \downarrow
\end{align*}
\]

☐ Embedded clausal structures occur when one or more additional clauses are embedded in a main clause, as in the sentence The boy that broke the window ran away, where the clause (The boy) broke the window is embedded in the main phrase The boy ran away. This aspect is also mirrored by the formal language.

\[
\begin{align*}
  b(bab)b & \rightarrow The \ boy \ (that \ broke \ the \ window) \ ran \ away \\
  \downarrow & \hspace{1cm} \downarrow
\end{align*}
\]

The proof that an FSA is limited in its ability to parse some natural language sentences may initially appear to be of little consequence for cognitive scientists, but Bever et al. (1968) argue convincingly that a FSA's computational mechanism is based on associative principles. The shortcomings of this machine are thus also shortcomings of associative devices, and of associative theories of language processing, more generally. Associative theories are typically based on the idea that the interpretation of a sentence involves a linear chain of left-to-right associations, with sentence meaning being just a combination of such associations, and this is precisely the processing mechanism underlying an FSA. Hence the message is clear; associationism (and this includes associative psychological theories such as behaviourism) is not powerful enough to deal with some facets of language and therefore fails as an adequate model of human cognition and language interpretation.
3.1.3.2 The functionalist position

The physical properties of a TM, provide some general constraints on the way in which it tackles an information processing problem (e.g. it is limited to processing a single symbol at a time). However, a physical description of a computational device (like a TM) has some drawbacks, because physical descriptions do not capture underlying information processing commonalities. Physically different systems may still be very similar from a computational or information processing perspective. For instance, a human player and a computer program may exhibit similar chess playing skills, but these obviously derive from very different underlying physical structures. Likewise, Babbage’s analytical engine is a computing device, but bears very little resemblance to any modern computer because it is constructed from gears and levers instead of micro-electronic elements (Pylyshyn, 1984, p. 57). Trying to explain the similarity in computational ability between these systems in terms of physical aspects is a rather hopeless task. Such cases where qualitatively different physical mechanisms instantiate the same information processing system, exemplify what has become known as the problem of “multiple realisability” (Dawson, 1998, p. 26). The problem serves to accentuate the fact that there is a class of things, to wit information processing devices, that are more appropriately and meaningfully described and analysed in terms of their functional properties and causal relations than in terms of their physical construction.

If we accept that cognitive systems are information processing devices, then it is plausible to conjecture that they too may be best conceptualised in terms of functional attributes. Putnam (1960) calls the latter position “functionalism”. Loosely speaking, functionalism is the view that the various states of an information processing system can be understood in terms of their functional or causal role with respect to the input of the system, the system’s output and the other states within the system causally connected to the states to be explained. Likewise, functional analysis is a type of explanation in which some system is decomposed into its component parts and the workings of the system are explained in terms of the capacities of the parts and the way the parts are integrated with one another. The component parts and their interactions are specified entirely in terms of causal relations. That is, the parts are picked out by their causal role in an input-output stream, and they display that role by virtue of their causal or functional connections to other surrounding component parts (Block, 1980; Dennet, 1971). Applied to cognition, the main thrust of the functionalist thesis is that a mental or cognitive state is a functional state of the brain, so that what matters for the clarification of such a state is not physical properties per se, but a pattern of causal relations, the relation between the state and other mental and physical states and its intentional properties (i.e. what the state is about). The goals and beliefs of an agent are such intentional constructs.
The idea is that what mental states should be viewed as functional relations between an agent and the world, and these states may be realised in different physical systems. An implication of this functional stance is that although the mind is implemented in the brain, it could in principle also be implemented in other machines with adequate computing power - such as a digital computer or a Turing machine (see Putnam, 1960).

Part of the rationale for the functionalist perspective is that it appears to avoid what Stillings et al. (1997, p. 370) call “Ryle’s regress”. Ryle (1949) reacted to intellectualist or rationalist theories which ascribe intelligent behaviour to the conscious application of mental rules. He argued that such intelligent agents are trapped in an infinite regress:

“... whenever an agent does anything intelligently, his act is preceded and steered by another internal act of considering a regulative proposition appropriate to his practical problem...Must we then say that for the hero’s reflections how to act to be intelligent he must first reflect how best to reflect how to act?” (Ryle, 1949, p. 31).

The problem is that when cognitive or functional terms are used to explain cognitive phenomena, these terms often assume some intelligence on their part, which must also be explained. This may lead to an infinite regress of appeals for intelligence to account for intelligence. Some theorists argue that the computational perspective overcomes this problem partly because it is based on the example of actual systems, such as digital computers, which arguably are not subject to such a regress (e.g. Stillings et al. 1997). A computer is endowed with a set of basic information processing components involving symbol manipulation operations such as searching for a symbol, concatenating symbols to form a complex expression, or moving symbols, as well as primitive representational units over which these operations are defined (i.e. different data types). These operators and units are part of the system’s functional architecture; they are invoked as primitives in particular information processing tasks, and need not involve any further decomposition. Dawson (1998, p. 160) contends that cognitive systems can likewise be analysed in terms of a set of basic processing elements, arguing that an explanation of these basic elements does not require further decomposition, but an account of how they are built into the cognitive architecture, and a detailed account of this architecture.

To recapitulate. The functionalist view is founded on the notions that cognitive processes are characterised by their functional properties and patterns of causal relations, and that these aspects exist and can be clarified on a specific cognitive-computational level of abstraction, independent of physical or implementation considerations. According to this view, the essence of an information processing system
is a collection of symbolic structures and symbol manipulators that decompose functionally, and whose functions and components are characterised intentionally (i.e. by their ability to represent entities in the world). Classical cognitive science is an attempt to understand the mind as just such a system.

### 3.2 THE CLASSICAL VIEW OF COGNITIVE ARCHITECTURE

The mainstream or classical approach draws from the abstract computational framework described in the previous section. Proponents of the classical approach typically adopt a functionalist perspective in which the emphasis is on clarifying functional, cognitive-level phenomena, and where issues relating to neural structures are often discarded as mere implementation concerns (e.g. Fodor & Pylyshyn, 1988). Classical theorists further postulate that any formalism must have the necessary expressive or “computational” power to account for aspects of higher cognitive functions. This postulate actually includes two claims: Firstly that the theory and its attendant formalism should be sufficiently expressive to cover all aspects of the domain (i.e. cognition). Secondly that higher cognitive functions such as language and reasoning provide a “litmus” test for any formalism. This notion of computational sufficiency is implicit in much of the early development of the classical approach. For instance, as shown in the previous chapter Rosenblatt’s perceptron was discarded because it proved to be limited in the range of functions that it could process.

In what follows the main assumptions and postulates associated with the classical position are set out. It should be noted, though, that this is a somewhat idealised version of this approach, and that many researchers working within the cognitive framework do not necessarily consciously situate themselves with respect to any specific philosophical position or set of foundational assumptions. Nevertheless, the approach is generally taken to have gained the status of an orthodoxy in cognitive science (cf. Johnson-Laird, 1988, p. 37-53).

The theoretical foundations of this classical view and the cognitive architecture that it implies are elaborated in works such as Fodor (1975) and Pylyshyn (1984), while more recent versions are given in Horst (1996) and Newell (1990). The primary assumption is that the mind is a universal computing device like a Turing Machine. Classical theorists posit a specific level of computational description, concerned with the functional architecture, the set of information processing components that make up the mind. The main intuition is expressed in Newell and Simon’s (1976) physical symbol hypothesis which states that a symbolic processing system constitutes “the necessary and sufficient” means for intelligent behaviour (Newell & Simon, 1976, in Haugeland, 1985, p. 40).
A physical symbol system is a computational system built up out of the following components: a memory, a set of operators, a controller, receptors (connecting it to the world), and a motor output. Its functioning resembles a UTM, in that the controller calls one of the operators which then processes the symbols stored in memory, just like a UTM consults the machine table associated with a specific TM. However, it is far more efficient system than a UTM because (a) its memory can contain structured symbolic expressions (e.g. sentences), and (b) it is endowed with a random access memory. Newell (1980, p. 151-154) proves that a physical symbol system can simulate any specific TM and is therefore equal in power to a UTM. Any general purpose computer constitutes such a physical symbol system, but so does the human mind according to Newell and Simon (1976). In fact, they contend that the ability to manipulate symbols is the scientific essence of thought and intelligence and a central property of any system that exhibits general intelligence (Newell and Simon, 1976, p.42).

### 3.2.1 Core postulates

The basic assumption that cognition is a form of symbol processing, gives rise to a number of specific claims about the nature of cognitive architecture (where by the term architecture is intended the structural components or constituent processes making up the mind). Some of the more salient of these postulates are listed below:

- The constituent elements of mental states are symbols which can be combined to form symbolic structures or expressions. These structures have representational or semantic content in that they denote entities or states of affairs in the world, and their interpretations are a systematic function of the semantic interpretations of their elements;

- Cognition is information processing. Cognitive activity is analysed in terms of an amalgam of processes such as inference, problem solving, search, and hypothesis testing. These processes are essentially formal, syntactic in nature, and involve manipulations of symbols;

- In terms of the TM model of computation, information processing entails the following of a sequence of rules or instructions. Hence, if cognition is conceptualised as an information processing activity, it follows that any given cognitive phenomenon is an instantiation of a set of underlying cognitive rules or procedures, and is explainable in terms of such rules;

- There is a sharp distinction between symbols and the processor rules used to manipulate them. A
description of a cognitive system is a two-pronged affair, entailing an explanation of the nature of the symbolic structures that represent knowledge, and a description of the nature of the rules and procedures used to manipulate them.

The classical position posits a level of description which focuses on the symbolic, information processing characteristics of the cognitive "virtual machine" (i.e. a machine implemented in software), yielding a picture of the mind founded on the idea that cognition is an information processing activity, and that thinking consists in the rule-governed manipulation of structured symbols (Lachman, Lachman & Butterfield 1979; Pylyshyn, 1984). As noted above, an implication of this assumption is that a qualitative distinction can be drawn between two components of an information processing system: structures or symbols that represent information and processes or rules that are used to manipulate symbols, solve problems and process information. These two components are reflected in the classical approach in two related theoretical constructs: the computational theory of mind (CTM), and the representational theory of mind (RTM)(see Macdonald & Macdonald, 1995, p. 5).

3.2.2 The computational stance

The CTM states essentially that the mind can be envisioned as a computational system. This theory draws from a computer metaphor and attempts to clarify psychological phenomena with the aid of concepts borrowed from computer theory. In fact, the CTM comes in two related flavours. In the first, the use of the computer metaphor does not commit the CTM to the claim that "the mind is nothing more than a computer" (which may or may not be true). Instead, it posits that the mind is sufficiently like a computer that the use of the metaphor, and the jargon associated with it, provides a set of concepts for theorising about thinking processes. On the second, stronger view, the mind instantiates a computer, it simply is a physically realised computational system. On both accounts, cognition is conceptualised as an information processing activity involving the manipulation of symbolic, mental representations in accordance with a set of rules or instructions that describe the phenomenon in question. For example, given performance data on psychological tasks a series of information processing events can be postulated that accounts for the performance. These processes are assumed to involve the manipulation of symbolic representations encoded in the mind, and are thus defined over mentally represented knowledge.

According to the CTM mental processes are computational in that they are governed by the formal syntactic properties of representations. As Fodor (1994, p. 8) puts it, mental processes are "mappings from symbols to symbols", and the causal relations holding between them preserve semantic relations. The
latter property has two related interpretations:

- Firstly, that cognition can be described in formal terms, as a computational mechanism that draws inferences on the basis of purely formal, logical relations between symbolic entities or propositions. Thus given A and the propositions A ⊨ B (where ⊨ denotes entailment), the truth of B can be inferred;

- Secondly, that there is a systematic mapping from symbols to the object they represent in that semantic relations between objects are reflected in their formal, syntactic relations, and in their causal connections.

The computer metaphor shows us how the human brain, by operating only on syntactic features of physical processes could be a semantic system. Computers execute algorithms (effective procedures) which are formal specifications of how representations must be manipulated to achieve a given result. An algorithm, in turn, is a sequence of purely formal, syntactic instructions according to which representations are manipulated, but which yields semantically interpretable results. Since algorithms can be carried out without any higher knowledge of their meanings they can be executed by physical systems (biological or engineered). For instance, a TM deals with the task of language recognition in a purely formal way, yet the resultant classification would nevertheless have semantic effects, because it separates meaningful sentences (i.e. sentences consistent with the grammar) from semantically problematic sentences (i.e. those that violate the grammar). The CTM draws from these insights, and attempts to account for cognitive processes by analysing them as formal, rule-based operations and by positing representational relations between symbols and objects or states of affair in the world. Hence, the basic assumption is that an agent produces meaningful behaviour by performing computations on symbolic structures that bear a representational relationship to the world, and that cognition can be conceived of as a kind of mental logic, a “proof theory” defined over internal data structures (Chater & Oaksford, 1991).

Classical symbolic models of computation are closely related to logical systems (Turing’s concept of a TM evolved from his work in mathematical logic). The close affinity to logic is reflected in the assumption that the mind instantiates some sort of logical calculus which represents information in a propositional, digital rather than an analog format, and relies on logic-based deductive mechanisms (see e.g. Johnson-Laird, 1983; Cann, 1993). Proponents of the CTM further posit that the mind exemplifies two general properties associated with logical systems: compositionality and variable binding.

- Compositionality means that the system must be capable of combining elementary constituents
to form more complex expressions. Natural language is compositional in this sense, because it allows words to be combined in new ways to generate novel sentences. Recall that Newell and Simon's (1972) notion of a physical symbol system, is likewise compositional, because it can form complex expressions by combining symbols (Section 3.3 above);

☐ Variable binding is a powerful means for generalisation and for encoding knowledge in terms of schemata. To illustrate: consider a statement in first order logic such as $\forall x \ [\text{Dog}(x) \Rightarrow \text{Barks}(x) \land \text{Has\_tail}(x)]$, which is to be read that everything (all $x$'s) that is a dog, barks and has a tail. If we are now told that Rover is a dog, we automatically infer that it barks and has a tail by binding "Rover" to the variable $x$. As will be discussed later, connectionist systems have been criticised because they do not present such a mechanism for variable binding.

The CTM thus describes the mind as a computational system operating on the basis of logical principles. It emphasis the syntactic, rule based aspects of computation and invokes these in the explanation of cognitive processes. The CTM is essentially concerned with syntactic processes, and construes cognition as a formal, symbol manipulation operation, so that an account is needed of cognitive content, of how semantics enters the cognitive system. This part of the classical approach is supplied by a theory of representation, which is the topic of the next section.

### 3.2.3 The representational thesis

The representational theory of mind (RTM) is concerned with the content of elementary mental constructs (i.e. symbols according to the classical perspective), and posits that cognition involves a construal of reality, a conceptualisation of the (real and imaginary) world with the aid of mental constructs (representations). The main idea is that symbolic mental tokens form the basis of conceptual structure, and that these represent individual objects or properties, while combinations of such tokens represent states of affairs. These mental tokens or representations are defined in terms of three properties:

☐ they are symbolic constituents and thus have individual semantic status, equivalent to that of a single word or concept;

☐ they are denotational in that they refer to entities; and

☐ they have a combinatorial power in that they can be combined into more complex units (e.g. expressions).
The RTM posits that knowledge is embodied in such elementary units of information, and that thinking and reasoning can be envisaged as the application of computational processes over such symbolic entities. Hence on this account, the mind is a representational system, composed of individual symbolic constituents (see Fodor & Pylyshyn, 1988).

The theory attempts to furnish an account of content, of the informational value of mental constructs, and of the relationship of such constructs to one another. On the most general level, the thesis has its basis in phenomenological experience (Shanon, 1993, p.7). Mental constructs such as visual imagery, memories and verbal recollections form part of everyday experience, and seem to suggest an inner symbolic system for coding experience. The implication here is that the occurrence of such commonplace psychological experiences and events can be explained by postulating a conceptual, representational system. This ‘naive’ stance has a more elaborate, theoretical counterpart that Shanon (1993, p. 8) refers to as the “epistemic rationale”, which is based on the argument that for behaviour to be meaningful it must be grounded in a (conceptual) substrate that carries meaning. In this epistemic sense, the RTM accounts for knowledge (and language) acquisition by postulating a system in which content knowledge is represented.

Shanon (1993, p. 8) cites the related argument that the knowledge that speakers have of word meaning in everyday conversation implies that they must have some or other conceptual representation in which the semantic properties of words are specified. Unless such an internal store of meanings is postulated, it would be difficult to explain the common observation that people can interpret a sentence in which they know the individual word meanings but not a sentence containing unfamiliar words. Evidently our knowledge of language is represented in some way in our mind. Moreover, there is considerable evidence from cognitive neuropsychological research that in some cases brain injury may lead to selective impairment of lexical knowledge.

The basic intuition underlying the RTM is that human knowledge and its manifestation in language can be explained if we assume that there is an internal (mental) conceptual system that serves as a medium for knowledge representation. The best worked account of such an inner medium is Fodor’s (1975; 1998) language of thought hypothesis, which posits that thinking is conducted in a special inner language, often referred to as “mentalese”. According to Fodor (1975), mentalese is not an equivalent of the language we speak but logically prior to it, it is the medium in terms of which thoughts are expressed, and our capacity for mentalese is not something we acquire or learn, but an innate capacity of our brains. Fodor (1975; 1998) justifies the mentalese position using some rather counter-intuitive arguments. For instance he argues that most concepts cannot be learned but are innately specified.

To summarise: the primary goal of the representational thesis is to account for mental content by positing
conceptual structures that serve both to embody the semantics of concepts and to link these concepts to external objects (i.e. to the represented objects). Although this general approach is favoured by many practising cognitive scientists, there is considerable debate about the nature of this content and about the way in which symbolic representations are linked to their real-world counterparts. Some of these issues will be considered in Section 3.3.5 below.

3.2.4 The classical approach to cognitive modelling

As noted above, the concept of computation plays a central role in this classical approach, causing Dennet to refer to it as “high-church computationalism” (Dennet, 1998, p. 219). The foundational role of computation in the classical approach is expressed in a thesis of computational sufficiency, namely that the right kind of computational structure suffices for the possession of a mind, and also in a thesis of computational explanation, which states that computation provides a general framework for the explanation of cognitive processes and behaviour. Part of the appeal of this computational metaphor in psychology and cognitive science stems from the fact that it yields a way of understanding and explaining cognitive phenomena that is broadly consistent with the physicalist assumption of modern science. The classical theory situates cognitive systems within a general class of abstract automata (i.e. physical symbol systems) that are general-purpose computing devices. These systems are capable of seemingly intelligent behaviour, yet physically realisable and thus open to scientific investigation, because their computational components and functional architecture can be studied. Conceptualising the mind as such a system avoids some of the pitfalls of dualistic conceptions in terms of which mind and brain belong to completely different and irreconcilable categories. Of course the classical case should not be overstated. It is quite possible that some aspects of mind (e.g. consciousness) might eventually turn out to be incompatible with a physicalist and computational conception of mind (but see Chalmers, 1996).

The classical approach’s contribution to cognitive theory can be appreciated if it is viewed in the light of a general problem associated with the development of psychological explanations within the scientific framework. Scientific explanations are traditionally concerned with uncovering law-like regularities in phenomena and involve a process of subsumption under general laws, called the “deductive-nomological” method (Hempel, 1965; Kukla, 2002, pp. 83-88). The model is based on a “covering law” model, equating the explanation of phenomena with subsumption under projectible (from observed to unobserved instances), counterfactual-sustaining universal statements. A proper explanation, According to this view, consists of a set of statements in roughly the form of modus ponens (though the explanation may be probabilistic rather than deductive): a statement of the explanandum (B), a set of initial conditions (A), and
a law connecting the initial conditions with the explanandum (if A then [probably] B). For instance, mature sciences such as physics strive to uncover the principles underlying natural phenomena by a combination of assumptions, observations and deduction, and are concerned with the formulation of general laws (e.g. Newton’s second law $F = ma = dp/dt$). In the cognitive and behavioural sciences the discovery of such laws has been more problematic. There are some laws in psychology. Newell, (1990, p. 3-12) mentions Fitt’s law (which expresses the time to move a finger from one point to another as the logarithm of the ratio of the distance moved to the size of the target region), the well known power law of practice, and Newell and Rosenbloom’s (1981) formulation of the chunking mechanism. However, by and large psychological phenomena have proven to be resilient to such subsumption under general laws. In fact, in a now famous article, Davidson (1974) argued against the possibility of formulating strict psychological laws. He contended that a systematic indeterminacy is associated with phenomena such as the beliefs, attitudes and plans of an agent (which he regards as the domain of the mental), and consequently that psychology cannot give rise to a closed comprehensive theory conducive to the formulation of deterministic laws of behaviour. The best psychology can hope for are statistical generalisations, but these are different from the statistical laws operating in physics, for instance, for in psychology they are “irreducibly statistical correlations that resist, and resist in principle, improvement without limit” (Davidson, 1974, p. 42). From this Davidson concludes that there is a conceptual difference in the explanatory powers of psychology and physics, and that these are indeed different categories of science.

The computational approach yields a different methodology for constructing cognitive and psychological explanations that, at least on the face of it, appears to overcome Davidson’s (1974) pessimistic prognosis. In terms of this methodology, the concern is not with developing general laws, but with the construction of a mechanism that generates the required behaviour. Classical theorists study cognitive phenomena by constructing explanations that describe the underlying mechanism or competency associated with the phenomena. The explanatory strategy involves what Dawson (1998, p. 108) calls a "forward engineering approach" to the study of cognitive systems. In reverse engineering the physical properties of a system are studied in order to establish how the whole system works, whereas forward engineering entails the actual construction of the system in question. In the cognitive science framework, researchers are not trying to design a physical system, but to recreate the virtual machine that exhibits the relevant cognitive behaviour. In the classical case, this approach is typically reflected in two related aspects, firstly a decompositional strategy in terms of which a given complex phenomenon (e.g. memory) is described in terms of interacting subsystems (different memory processes) and secondly in a specification of the information processing attributes of each of the subsystems. These descriptions are often presented in purely verbal form, on the assumption that formalisation (if at all possible) can be pursued at a later stage.
There is also an implicit, ultimate objective of actually simulating the behaviour in a computer program or even expert system (Dawson, 1998). The approach thus tries to explain how actual instances of behaviour are generated, with the additional implicit requirement that the explanations must be constructive, that is effectively computable or realisable by a formal mechanism, e.g. a Turing machine. Hence, notion of mechanism in classical cognitive science ultimately rests on the concept of effective computation or Turing-machine realizability.

A paradigmatic example of this classical research strategy is Chomsky's (e.g. 1965) approach to language. He conceptualises language as an interacting set of subsystems, involving inter alia, a phonological system, concerned with the sound structure of language, a semantic system which specifies word meanings, and a syntactic system which governs the structural component of language. Each of these subsystems is described in terms of a set of representations (symbols or features) and rules that regulate the processing of the relevant information. In the more recent version of the theory, Chomsky (1981; 1993) amended his approach to allow for a more abstract description of rules in terms of principles and parameters (which are adjusted for individual languages) but the gist of the approach remains the same. It focuses on a computational description of language as a symbolic system, and the task of the researchers is seen as that of analysing the functional components of the system and describing their operation. This approach shifted the emphasis from the study of overt, manifested verbal behaviour (prevalent in earlier approaches in linguistics and psychology) to the study of the cognitive underpinnings of such behaviour and thus to the study of cognitive mechanisms. It is marked by an abstract, formal description of the syntactic rules (and subsequently principles and parameters) associated with linguistic competence (i.e. linguistic knowledge) rather than with behavioural aspects such as language performance. The distinctly classical nature of the approach is that the mechanism is construed in terms of rules and symbolic constituents that are operated on by a cognitive virtual machine.

Dennet (1991) refers somewhat caustically to this classical conceptualisation of the mind as a search for the 'Master Program', the implication being that classical theorists assume that the mind is governed by a central program, a single logically sound system of explicit rules and representations. There are many problems with the classical research strategy, such as how does one carve up a complicated behaviour into an organised set of functions and how does one validate a functional decomposition? But despite these issues, the approach has proven to be very fruitful in at least some domains of cognition. For instance, a significant contribution of the classical view is that it accounts for the creativity and systematicity of thought (Fodor & Pylyshyn, 1988). It accounts for creativity because the computational mechanism shows how cognitive representations can be productively combined to yield new structures, and it explains the systematicity of thought because the syntactic mode of processing invokes constituent structure (i.e. a
hierarchical organisation of information).

To summarise: Classical cognitive science is founded on the assumptions that mental structures are representational, that cognition is a form of information processing, and that cognitive phenomena can be explained with the aid of information processing concepts. On the classical view, cognitive processes are assumed to be essentially syntactic, symbol manipulation operations (e.g. searching for, moving and substituting symbols) and the representations are assumed to be build up out of discrete, word-like elements, which have referential content. This yields a concept of mind as some sort of abstract automaton, a general-purpose computing device, which employs rules to direct the transformation of underlying mental representations (symbols). Theorising in the classical sense mainly exploits a forward engineering strategy in terms of which complex cognitive phenomena are analysed in terms of interacting subsystems and explanation is aimed at clarifying the effective procedure (i.e. algorithm) needed to reproduce the underlying competency associated with the phenomenon. This type of research is typically pursued both at a fairly broad level of theorising, where the explanations are mostly couched in verbal terms, and in finer grained theories in which techniques such as formal analysis and computational modelling are brought into play.

3.2.5 Problems for the classical approach

The classical approach is still reasonably well-ensconced in mainstream psychology and the other cognitive sciences, but nonetheless has many critics. Many of the assumptions it makes about processing and representation are controversial, and the very notion of information processing has been challenged on various grounds. In fact, some critics have challenged the assumption that the concept of computation is relevant (i.e. either necessary or sufficient) to characterise the functioning of the mind. These objections will be dealt with in some detail in Chapter 6, as some of them apply to the connectionist framework as well. For now, it suffices to just briefly mention some of the points of contention.

Firstly it has been argued that computationalism has difficulty in accounting for the vast amount of background information that people make use of when drawing inferences, and hence that computational explanations are not sufficient for explaining cognition (Dreyfus, 1974). This issue surfaces specifically in the case of classical logic-based explanations of processes such as commonsense reasoning presented in both AI and psychology. For example, first-order predicate logic is monotonic (the axioms monotonically build on one another) whereas human reasoning is nonmonotonic which means that a speaker can change his or her mind about a fact previously accepted as true when new information comes
to light that is inconsistent with his or her previous assumptions. Various formalisms have been proposed to account for nonmonotonicity (Reiter, 1980; McCarthy, 1980), but all the logic-based formalisms developed so far appear to be intractable in the worst case because the theorem proving algorithms underlying them are formally undecidable, and inefficient in practice, often requiring an exponential number of logical steps to ensure that the premises remain consistent (see Oaksford & Chater, 1991, p. 2; Russell & Norvig, 1995, p. 460). Thus unless an efficient theorem prover is found for the nonmonotonic logics - a possibility that seems rather remote at present, (but see Cholewinski, Marek, Mikitiuk & Truszczynski, 1999) - the current systems will fail to provide a tractable mechanism of commonsense reasoning.

Secondly, some researchers claim that the classical approach has made little progress in advancing our understanding of the part of cognition devoted to general reasoning, which Fodor (1983) calls the "central systems". Fodor even doubts that a (classical) theory of central systems is possible, because such systems have two properties that render computational analyses problematic:

- They are "isotropic" which means that the facts taken to be relevant to a specific act of reasoning may be drawn from anywhere in the content of the whole knowledge base, that is, anything is potentially relevant to anything else; and
- They are "Quineian" which means that the degree of certainty attached to any belief or inference drawn depends on properties of the entire belief system.

Thirdly, some researchers point out that the computer metaphor has difficulty in accommodating the notion of content and meaning, and consequently that a description of a computational mechanism typically leaves aspects relating to content and meaning unanswered. Computational mechanisms involve the manipulation of symbol tokens, but meaning does not come for free and an explanation of how these symbols actually manage to convey meaning is required. The RTM is supposed to furnish such an account, but the theory simply posits that symbols have referential attributes without explaining how their content is grounded in external objects (see Lakoff, 1987). The problem of content is also illustrated in Searle's (1980) well known Chinese room gedanken experiment, which will be considered in Chapter 6.

Fourthly some theorists argue that some aspects of human thinking may not be computable, and hence that computationalism would not be able to describe these aspects. For instance, Penrose (1995) maintains that mathematical insight is not computable, and invokes Gödel's incompleteness theorem to make his point.
Fifthly, classical theorists are essentially preoccupied with abstract computational issues, and have largely ignored the question of how such computational processes are implemented in the human brain. The resultant theory has little to say about the way in which such algorithms are realised in the neural hardware, so that the theory creates a schism between the neural and cognitive levels of explanation.

Finally, classical theorists have been primarily concerned with issues pertaining to the representation and processing of information, but have paid scant attention to issues of learning. For instance, the typical description of a TM does not cater for learning. Some classical theorists such as Chomsky (1980) and Fodor (1980) adopt an extreme rationalist position, and go so far as to assert that in some domains such as language (Chomsky, 1980) and concepts (Fodor, 1975), humans are endowed with predetermined genetic structures and that little or no learning takes place.

It is in the wake of these problems with the classical position, that neural network modelling has emerged as a slightly different conception of cognitive architecture and the attendant processing mechanism. In contrast to the classical position, neural network researchers do take the interaction between cognitive and neural levels seriously, and attempt to foster insight into the issue of human learning. We now turn to consider this approach in more detail.

### 3.3. The Neural Network Paradigm and Computation

The classical approach is an influential and productive paradigm, both in terms of the amount of research generated (see e.g. Stillings et al., 1995) and in the light of its strong theoretical basis (e.g. in computability theory). Thus, although there are problems associated with application of the computer metaphor to cognition (as briefly discussed in Section 3.3.5.), the hypothesis has proven quite resilient and is still endorsed by many cognitive researchers (see e.g. Fodor, 1998; Chalmers, 1995). The approach has been able to withstand criticism because many researcher believe that there is no really viable alternative metaphor for conceptualising the working of the mind, inducing Fodor (1975, p. 27) to remark that the “only psychological models of cognitive processes that seem even remotely plausible, represent such processes as computational”. This remark should be seen in the light of a fairly pervasive intuition by computationally oriented psychologists, philosophers and other cognitive scientists, that the available alternative theoretical options are either too vague (e.g. introspective accounts) or are not sufficiently powerful to account for higher-level cognition (e.g. associative approaches)

With the resurgence of connectionism in cognitive science it becomes pertinent to ask how it fits into the
classical scheme, and whether connectionism might offer a feasible alternative to the classical paradigm. This question is far from easy to answer, as will be shown below. To pre-empt the discussion, the answer to the question hinges on two issues; (a) whether connectionism is really a different paradigm or just a slight re-interpretation of the classical account (e.g. a classical approach clothed in parallel processing machinery), and (b) assuming that it is indeed a different approach, whether it is powerful enough to function as a theory of cognition, and particularly of higher level cognitive phenomena.

3.3.1 Conceptualising cognition in terms of neurocomputational principles

A first problem in relating connectionism to the classical position, is that connectionism is far from a unified paradigm. As the discussion in the previous chapter served to illustrate, there are a variety of different approaches in connectionism, and these exploit different network architectures and learning algorithms. Moreover, there are substantial differences in opinion about the interpretation of connectionist networks, and particularly about their status as psychological models. There are even fundamental differences in opinion about whether connectionist models should be regarded as computational models at all. Certain models such as recurrent or attractor nets implement dynamical systems, leading some researchers to argue that connectionist models should be analysed in terms of a dynamical systems perspective, rather than a conventional information processing paradigm (e.g. Churchland, 1995; van Gelder, 1998). The latter issue will be addressed in Chapter 6, but for the time being, it is useful to adopt as a working hypothesis the assumption that connectionist systems are computational in the general sense, because they perform input-output mappings - incidentally, this is also the view presented in the PDP books (see e.g. Rumelhart et al., 1986, p. 10-12) - but that the type of computation differs from classical computation in some important respects, because it is based on a notion of "neurocomputational", or "parallel distributed" processing (Hecht-Nielsen, 1990). The neurocomputational approach was presented in the previous chapter.

There is no doubt that connectionist models have a different feel than standard symbol-processing models. The units, the network topology and weighted connections, the functions by which activation levels are transformed in units and connections, and the learning (i.e. weight-adjustment) function are all that is "in" these models; one cannot easily point to rules, algorithms, expressions, and the like inside them. In fact there are significant differences between the way in which these models process information and the classical, symbolic conception of information processing. Some points on which these approaches diverge, are listed below:

87
Whereas classical systems apply inference rules to symbolic variables, ANNs apply evolutive principles to numerical variables. Instead of calculating a solution, the network settles into a condition that satisfies the constraints imposed on it. The processing of these models is more appropriately interpreted as a form of constraint satisfaction, in terms of which a network tries to reconcile the constraints imposed by large collections of small processing units, than in terms of a process of rule following in the classical sense.

Classical systems are based on a serial architecture, and typically include a main function that controls the flow of information. ANNs have no central control in this classical sense. Processing is distributed over the network and the roles of the various components change dynamically. Some part of the network may develop a regulatory function depending on the evolutionary needs of the system, but they are not fixed by some prior programming decision.

Every symbol in a rule-based system has a precise, predefined meaning, that may be called its local representation. In ANNs individual units have no predefined meaning. Instead it is typically assumed that meaning resides in changing patterns of activity ranging over groups of units in the network. In a neural net, information is therefore represented in a somewhat fuzzy, distributed fashion involving large collections of units in the network, unlike the crisp symbolic entities that are assumed to represent information in classical systems.

Computational processes in classical systems have well-defined terminal conditions and results are only produced when these conditions are met. ANNs tend to dynamically converge on a solution, usually in an asymptotic fashion. Moreover, the process need not terminate, and is usually not associated with a single possible outcome.

The internal structure of an ANN develops through a process of self-organisation, whereas rule-based systems rely on preprogrammed instructions that define their internal structure largely in an a priori fashion. In this sense, learning is an implicit characteristic of neural networks. In rule-based systems learning only takes place through explicitly formulated procedures.

In symbol systems there is a crisp distinction between two levels: that of the symbols and that of the rules—data and program. In ANNs there is only one level: units and connectionist weights so that the distinction between memory and processing becomes blurred. All the information that a network has access to (i.e. its memory) is stored in patterns of activation in the network, and these same activation patterns instantiate its processing mechanism.
As the above points illustrate, ANN models differ in various respects from their symbolic counterparts. On the classical account, information is represented by strings of symbols, just as we represent data in computer memory or on pieces of paper. The neural network researcher claims, on the other hand, that information is stored non-symbolically in the weights, or connection strengths, between the units of a neural net. The classicist believes that cognition resembles digital processing, where strings are produced in sequence according to the instructions of a (symbolic) program. The neural net modeller views mental processing as the dynamic and graded evolution of activity in a neural net, each unit's activation depending on the connection strengths and activity of its neighbours, according to the activation function. The basic concepts of a neural net, its topology, learning mechanism, activation functions and units do not have a clear parallel in classical systems.

**3.3.2 The shape of the debate**

On the face of it, these approaches may seem very different, because they instantiate different styles of computation. However, some theorists do not regard connectionist theories as a challenge to classicism, but seek an accommodation between the two approaches. They hold that the brain is indeed a neural net, but also a symbolic processor at a higher and more abstract level of description. So the role for neural network research, according to these theorists, is to discover how the machinery needed for symbolic processing can be constructed from neural network materials, so that classical processing can be related to the connectionist account (Fodor & Pylyshyn, 1988).

According to this view, neural net models occupy an intermediate level between symbol processing and neural hardware: they characterize the elementary information processes at a slightly lower level that is closer to the neural level, but they do not really bring anything new to cognitive theory, and cognitive phenomena can be coherently, and accurately described at a more abstract level by adverting to the concept of a virtual, symbolic machine that performs information processing operations in the classical sense. These theorists accept that progress in connectionist modelling might eventually force some revision of symbolic models, because traditional assumptions about primitive mechanisms are likely to be neurally implausible, and complex chains of symbol manipulations may be found to flow naturally from the primitive computational powers of neural networks. Nonetheless, in this scenario there is a well-defined division between rule-based descriptions and their realisation in connectionist systems, each playing an indispensable role in the explanation of a cognitive process. The proposed scenario is one where most existing types of symbol-processing models would remain mostly intact, and, to the extent they have empirical support and explanatory power, actually dictate many fundamental aspects of network.
organization. (see, Touretzky, 1986, and Hinton & Touretzky, 1985, where neural networks implement aspects of LISP and production systems, respectively). The main concern of connectionism, on this account, is that of implementation; of suggesting details about how symbolic, classical-level theories are realised in the neural wetware.

The implication of this implementation interpretation is that connectionist theories are simply diluted products, derivatives of their classical counterparts. However, there is reason to resist these negative connotations, because there is a sense in which connectionist theories go beyond classical explanations. For instance, connectionist systems incorporate a learning mechanism which flows naturally from the computational structure, whereas traditional models of computation in the classical context, such as the Turing machine model of mind, largely ignore the issue of learning. Clearly there is some sense in which brains are capable of learning, and adapting to their environment, and this is explained by connectionist rather than classical theories. Furthermore, even if connectionism only offers an implementation account, this is no mean contribution to cognitive theory. Classical theorists accept that physical and computational levels are related and that symbolic systems are ultimately implemented in the brain. However, no coherent theory about the physical underpinnings of cognitive processes have been forthcoming from the classical camp, and the approach has so far only presented the “Total Mystery” as its answer (Smolensky, 1995, p. 226). The fact that connectionist systems are endowed with an architecture that is at least neurally plausible at an abstract level of analysis therefore helps to bridge the chasm between cognitive and neural explanations.

Prima facie there is thus some theoretical justification for neural network research. But there is also reason to believe that neural network theories may actually transcend classical explanations. In the implementation view, connectionism is treated as a lower level enterprise that explores the possibilities for implementing classical models with parallel algorithms and machines. A more radical alternative is to begin with the basic concepts of connectionism and use the specific, idiosyncratic properties of networks to build a new theory of cognitive architecture. Such a theory would be a direct challenge to, rather than simply an extension of, classical models. In this more radical view, symbolic processing reflects an incorrect assumption about how the mind works, because it fails to capture some aspects of conceptual structure, such as the fluid nature of concepts and word meanings, which often change dynamically with context. There is therefore a sense in which neural network modelling might offer a more complete cognitive theory than the classical version. The next chapter explores this relationship between neural network models and classical conceptions of cognition in more depth.
A SUBSYMBOLIC APPROACH TO MODELLING CONCEPTUAL STRUCTURE

The main purpose of the previous chapter was to describe the information processing approach in psychology and cognitive science, and to situate it in the foundational framework of the theory of computing, and specifically in the context of the notion of a Turing Machine as an abstract model of computation. In the chapter the main theoretical postulates and assumptions underlying the classical theory were also presented. The core assumption of this classical view is that cognitive processes involve formal, syntactic operations defined over discrete semantic structures, so that the mind can be conceptualised as a rule-based, 'algorithmic' system operating on symbolic constituents.

It was suggested in the previous chapter that connectionist models may be distinguished from their classical counterparts because they rely on a different style of processing (the spreading of activation values rather than a process of rule-following), and because they draw from a different way of representing information, typically a distributed representational scheme instead of the system of local, symbolic constituents adopted in classical models. These different assumptions about the processing and representation of information also have implications for the style of explanation adopted in these two approaches (this will be discussed in more detail later in this chapter), so that there is reason to regard connectionism as a new theoretical option in cognitive science, which may even be incompatible with the classical thesis. However, in the previous chapter, it was pointed out that although there are seemingly significant differences between these approaches, the possibility that connectionism might offer a radically new alternative for cognitive theorising has been resisted by some researchers who claim that it should rather be construed as theory of how classical, symbolic theories are implemented in the neural system. According to this view connectionist models are pitched at a different level of explanation (i.e. a level concerned with implementational issues) and do not compete with classical theories, but show how classical theories can be realised in neural-like hardware.

This chapter further explores the relationship between classical and connectionist models, concentrating mainly on models and approaches to memory and learning developed under these two paradigms. Any model of cognition, and any artificial intelligence system that attempts to model aspects of human cognition, is forced to present an answer to the puzzle of how knowledge is stored in the brain. As intimated above, connectionist and classical researchers evidently adopt somewhat different formalisations of knowledge, and offer different conceptualisations of the way in which knowledge is stored. However, although the two approaches diverge in some respect, it is not clear whether these
differences are really significant at the level of cognitive theory. The crux of the issue is whether the connectionist and classicist viewpoints can be collapsed into one another if considered from a slightly more abstract and fundamental level of theorising, and indeed (more importantly) what the appropriate level of abstraction is in terms of which cognitive phenomena should be studied. These are the issues I begin to grapple with in this chapter. The present chapter has three main aims: (i) to reassess connectionism in the light of the implementational critique, (ii) to illustrate the application of ANN models in the domain of memory theorising, and (iii) to present a conception of connectionism developed by Smolensky (1988) who responded to some aspects of the classical critique of connectionist research by formulating a “subconceptual” approach to the explanation of cognitive phenomena. The chapter paves the way for a more detailed discussion of the strengths, as well as some of the limitations associated with connectionism as an explanatory framework in psychology and cognitive science.

It should be noted right from the start that discussions of the relationship between classical and ANN conceptions of cognition have become associated with a number of rather murky philosophical problems (some of which will be introduced in the following two chapters). Moreover, connectionism is an evolving research programme, and ANN modellers differ quite significantly in terms of how they think about the role of representations in cognition. Hence, the perspective on the classicist versus connectionist controversy developed in this chapter is not intended to resolve it, but merely to put some of the main issues into relief.

4.1 MEMORY AS A DOMAIN FOR CONNECTIONIST MODELLING

Memory has been a popular domain for experimenting with ANN models since the beginning of the field of connectionist research. Many of these models implement a form of associative memory and therefore lend themselves naturally to simulating memory storage and retrieval, which cognitive psychologists have traditionally described by positing associative mechanisms. In addition, the conception of memory in terms of networks pervades both cognitive psychology research on memory and learning, and AI approaches to knowledge representation. For example, semantic networks were initially proposed by Quillian (1968) and further elaborated to model semantic memory retrieval as a form of spreading activation by Collins and Loftus (1975), and have since been used extensively to explain the hierarchical organisation of information in memory (e.g. Reisberg, 1997, pp. 50-75).

One of the reasons why ANN models have been applied to memory is because memory researchers have often conceived of memory representation and processing in terms of distributed systems, so that PDP-type connectionist networks constitute a natural computational platform for exploring such distributed formulations of memory. Moreover, ANN models are used to capture aspects of memory development, and therefore serve to integrate memory learning and representation whereas these are sometimes treated
as separate categories in information processing approaches. The connectionist literature on memory is vast, the examples given below are only illustrative of the type of studies that have been conducted in the connectionist framework.

O’Reilly and Rudy (2001) developed a model to show how rapid learning in the hippocampal system can be accomplished if a split network system is postulated, and showed that their model accounts for a wide range of data in animal learning. Burgess and Hitch (1992) developed a model of the articulatory loop in working memory. The model makes some of the kinds of recall errors found in previous research on people and incorporates mechanisms for serial order and sensitivity to temporal context. Farah and McClelland (1991) constructed a model of semantic memory that attempts to implement the distinction between visual and functional knowledge. The researchers lesioned the model by destroying some of the connections and units in the net in order to observe the differential effect that damaging the net has on knowledge of living and non-living things. The specific processing assumptions used in the model are similar to McClelland and Rumelhart’s (1985) model which will be discussed further down in this section.

Hinton and Shallice (1991) showed how damaging the semantic memory of a net designed to model reading processes produces the semantic errors of deep dyslexics. Similar to such patients the net would substitute semantically related words such as boat and yacht for one another while also making phonological substitutions. This pattern of errors has been difficult to reproduce with classical information processing models. Bairaktaris (1995) developed a model that simulates the temporal chunking of serially ordered input sequences. He designed a modular recurrent net which was trained with backpropagation through time (e.g. Elman, 1990). The model was aimed at exploring low-level processes implicated in activation maintenance and temporal synchronization of temporally distributed input.

As these examples serve to illustrate, connectionist models have been developed to simulate various aspects of memory performance and memory structure. Nevertheless, connectionist models have not been enthusiastically received by all researchers. One criticism levelled at connectionist approaches is that they merely provide parallel processing theories of cognitive models which can be more parsimoniously defined using classical vocabulary. In articulating this line of critique, researchers tend to invoke Occham’s Principle, commonly adopted in scientific explanation, which basically states that there is no need to entertain extreme complexity if a simple model will do the trick. However, before considering this criticism in more detail, it is useful to first flesh out some of the details associated with ANN models of memory. Thus, in order to show how an ANN can accomplish human-like (cognitive) information processing abilities, I will consider two connectionist simulations of memory processes. The simulations stem from the modern resurgence of connectionism during the 1980s, and are therefore

93
somewhat dated by present day standards in connectionist modelling. Nevertheless, they serve to exemplify some of the basic processing principles associated with connectionist nets. The first example derives from a study by McClelland (1981) and the second is a simulation of memory processing by Rumelhart and McClelland (1985). These two studies are dealt with separately.

### 4.2 ILLUSTRATING CONNECTIONIST MODELS OF MEMORY: TWO CASES STUDIES

As noted above, the two examples of ANN models presented here are somewhat simple, but have been chosen for expository purposes because they exemplify the connectionist approach to memory and representation. Several sophisticated ANN models of aspects of memory have have been proposed such as Small, Hart, Nguyen and Gordon (1995) and McClelland, McNaughton and O’Reilly (1995), but their complexity is of such a nature that they are not suitable for expository purpose. The advantage of simple models is that they make the salient principles associated with connectionist formulations of memory easily visible.

#### 4.2.1 Interactive memory: The Jets and Sharks simulation

McClelland (1981) presented a simulation which is intended to illustrate the operation of an interactive memory system. He invites us to imagine a group of somewhat unsavoury individuals belonging to two fictitious gangs, The Jets and The Sharks (the names are borrowed from the musical “West Side Story”) in which members of the gangs are classified in terms of five other attributes: name, profession, age, education, and marital status. The aim of the study is to model the way in which humans might abstract and represent information in connection with the two gangs. Presumably, this information is gradually assembled in the course of a number of episodes. For instance, Fred wants to sell you some white powder at a party, you overhear someone in a bar commenting that Don may have forged his school results to get into college, you read that Rick married a beautiful model but that the marriage did not last, and you’ve heard that Ken is a Shark, and have seen that Ken and Karl usually hang out together. On the basis of such information you deduce that Fred is a drug pusher, that Don has been to college, that Rick is divorced and that Ken and Karl are both Sharks.

McClelland’s (1981) interactive and activation model (IAC) is a simple, localist ANN that represents information about the two gangs. The Jets are mostly in their twenties, single and have only a Junior High School education (i.e. grade 10), but no single Jet has all these characteristics. In contrast the Sharks are older, married, and have a High School qualification (i.e. a grade 12, or ‘matric’ certificate). But as in the case of the Jets, no single Shark has all the prototypical Shark properties. Members of both gangs are equally likely to be pushers, bookies, or burglars by profession. In the model each individual is represented with his own connectionist processing unit, called an ‘instance unit’. The model also
contains one property unit for each property that an individual might possess. Names are treated as such properties, so that there are units to represent the name for each gang member, and similarly for gang membership, education, age, and marital status. Bidirectional, excitatory connections between units are used to link the instance units representing names to those representing their properties. The instance units formed a group or pool of units which are mutually inhibitory so that if one becomes active it will tend to suppress the others. Units for each type of property are likewise grouped into clusters of mutually inhibitory units.

The focus of the study was on the process of constructing representations of material not explicitly stored in memory, such as the construction of a composite memory of a typical Jet or Shark. In the model such a reconstruction is achieved by simply activating the unit for Jet, which will then send activation to the instance units for all the Jets and these in turn would pass activation on to the properties linked to each instance. The basic IAC network consists of processing units that are organized into competitive pools. Connections within pools are inhibitory; this produces competition within the pools as strong activations tend to drive down weaker activations within the same pool. Connections between pools, however, are normally excitatory and bi-directional resulting in a form of interactive processing. Units within the network take on continuous activation values between a minimum and a maximum, with their output normally equal to the activation value minus some threshold (although this can be set to zero without loss of generality).

The inhibition among the instance units prevents any of these units from becoming too strongly activated, but they all pass on some activation to their property units. If the properties of the typical Jet become active (age in 20s, single, JH education) all the occupations will be partially activated because they are equally distributed among the Jets. In contrast, if a network is only informed that a particular individual is a Jet, the typical properties of the Jets listed above, will automatically be activated. The IAC model therefore exemplifies a desirable property associated with ANN models of memory; they can spontaneously generalise from examples. The IAC model also illustrates how distortions can occur in a memory system. When the model is used to try to retrieve the properties of a single individual by activating the unit for that individual’s name, the individual’s properties become more strongly activated than any other individual, but as the activation process continues those of the other similar individuals also become partially activated. This occurs because the properties associated with the target individual transmit activation to the instance units for other individual’s, and these in turn tend to activate the property units connected to them.

This distortion can be illustrated by activating the instance unit of one gang member, George, and the property unit denoting his occupation. This resulted in the instance unit becoming active and activating the property units for Jet, 30s, College education, and single. Several other Jets have many of these
properties and they are all pushers. As a result the model will complete the activation pattern by filling in an occupation for George as pusher, as shown in Figure 4.1 below. In a sense George becomes guilty by association. The model thus illustrates how memory distortion can occur by virtue of a simple associative mechanism embodied in the weighted connections linking units in an ANN. These mechanisms are often beneficial because they allow the formation of generalisations over similar instances and the completion of missing properties based on the properties of other similar individuals, but can also result in incorrect generalisation, based on typical or default expectations.

![Figure 4.1](image.png)

A computer simulation of the interactive model. As the display shows, presenting ‘George’ as input to the IAC net during testing, results in the activation of ‘Jet’, ‘age in 30s’, college education, ‘single’, and ‘pusher’.

Notice that the model shows how preexisting knowledge, which often aids in the construction of memories, can sometimes seep into and corrupt new memories. Similar processes play a role in human memory as Schacter (2001) points out. For example, the reconstructive aspect of human memory is illustrated by Bartlett’s (1932) famous experiment in which the memory recall data of American college students suggest that they had reconstructed the content associated with an American Indian tale called the War of the Ghosts to fit in with their western schemata about waging war. The students organised the events of the story into a specific chronological order, whereas the actual story had no real beginning or end.

The next example illustrates how an ANN can acquire general and specific information about everyday objects for storage in a distributed memory system.
4.2.2 Rumelhart and McCleland’s distributed model of memory

The following example is an ‘attractor’ model (see Section 5.1.4 for a discussion) of memory presented in McClelland and Rumelhart (1985). The concept of an ‘attractor’, borrowed from dynamical systems theory (see Abraham, 1995; Amit, 1989), is useful for the description of memory for two reasons. Firstly it helps to convey the idea of memory as a non-deterministic system. Similar cues or ‘input information’ received from the environment will tend to evoke a particular memory or category in a conceptual structure. However, the triggering of a memory is a probabilistic matter. The more explicit the cues, the more likely the fact that the memory will be evoked, but there is no guarantee that the particular memory will be recalled. Secondly, it is consistent with everyday experience that memory recall is possible even with slightly degraded input information. Using the concept of a memory organised into attractors, one can argue that partial or noisy information may be sufficient to elicit recall if the input falls within the region (i.e. basin) of an attractor. Thus even if the description is somewhat abstract, thinking of memory in dynamical terms captures some aspects of human memory processes. In fact, Van Looecke (1991) showed that a connectionist model of the dynamics of concept learning fits some of the empirical data associated with prototype formation documented in studies by Rosch (1973; 1975). This compatibility of some ANN models with cognitive psychology research on memory and categorisation is also illustrated by the McClelland and Rumelhart (1985) study.

McClelland and Rumelhart’s (1985) attractor network was developed to simulate the acquisition and representation of information in ‘semantic memory’ (see Tulving, 1972; Eysenck & Keane, 2000, pp. 185-187). Their model of memory is inspired by psycholinguistics studies of children’s acquisition of everyday semantic categories such as dog, toy, or candy. During early development, a child might see many dogs of different types, but with overlapping details, and eventually abstracts a representation of the category on the basis of these examples. In an influential article, Rosch (1975) contended that the representation developed is a prototype of the category, and that the memory system is organised in terms of such prototypes. To model the process in terms of which prototypes are abstracted and represented in the memory system, McClelland and Rumelhart (1985) designed a two layered auto-associative network (i.e. a symmetric net with the same number of inputs and outputs). In their model the units were organised into memory modules, and each output unit had recurrent connections back to the other units, as illustrated in the Figure 4.2 on the next page.

The actual net consisted of 24 inputs of which the first 8 were used to represent the names of individual category members (e.g. different dog breeds), while the other 16 encoded salient visual attributes
associated with category members. It can be assumed that the names of dogs do not exhibit any underlying similarity, and these were therefore just different random patterns. However, dogs can be expected to exhibit some similarity in visual appearance, separating them from other adjacent animal categories, such as cats. To account for the visual similarity between different dogs, the visual part of the pattern (units 9-24 in the input patterns) was a distortion of a particular binary vector (+1-1 +1+1+1-1-1-1+1+1+1+1-1-1) which was taken to represent the appearance of a prototypical dog. To account for the varying appearances of individual dogs, the vector was distorted by randomly changing the sign of each of the 16 elements with a probability of 0.2.

**Figure 4.2.** A recurrent, auto-associative network receiving the pattern [+1 +1 -1 +1] as input.

McClelland and Rumelhart (1985) constructed 50 different patterns, each representing a different exemplar of the category. The weights in the auto-associator were initialised to random values with a mean of zero. The net was trained on these 50 patterns with the delta rule, and activity was allowed to cycle until the network settled into a solution (i.e. when the error curve reached asymptote). After training, the activations of the units in the name section were close to zero. Statistically, a weight in such an auto-associator represents a correlation between the unit and the corresponding elements in the input array, summed across all the patterns, so that the low correlations indicate that the net could not extract any similarities between the different names (of category members). In contrast, the weights in the visual section of the matrix reflected much more activity, suggesting that the net learned to associate specific visual attributes with the different categories in the training sample. Moreover, since the visual patterns were derived from a common prototype, the weights gradually began to take on the shape of the prototype pattern despite their initial random settings, and this happened despite the fact that the net had no direct experience with the prototype. This result echoes the results obtained in a well-known experiment by Posner and Keele (1968) in which subjects were presented with patterns generated by a
random distortion of a prototype pattern. When the subjects were later asked which pattern had been shown to them, they were as likely to judge that the prototype had been displayed previously as the patterns to which they had in fact been exposed. The network mirrored this aspect of prototype abstraction because given only a name, the network responded with a good approximation of the appropriate prototype visual pattern.

There are several aspects of the performance of this net that can be highlighted. First it is clear that the net is capable of storing and retrieving a number of non-orthogonal categories of patterns. Second, the network is capable of pattern completion so that it can respond appropriately to incomplete or partial patterns. This aspect is reflected in the net's ability to recall the name of a category given its visual characteristics, and its visual characteristics when presented with the category name. Hence, the net models a content addressable memory system because pertinent input directly activates the memory content instead of just the memory address as is the case in the conventional Von Neumann-type architectures. Third, the network abstracted a prototype of the category even though it was only presented with exemplars and had no direct exposure to the prototype. A further study (McClelland & Rumelhart, 1985; Rumelhart & McClelland, 1986) showed that the net learned some specific aspects of the exemplars themselves. During training the net encountered two particular exemplars much more frequently than the other patterns. It can be predicted that cases which occur frequently during the learning phase will leave a greater impression on the representation than cases encountered less frequently. This prediction was borne out by the simulation, because during testing the net typically presented the category prototype name as output, but in the case of the two repeated exemplars it responded with their specific names. Evidently, the salience of the two repeated patterns produced an independent representation of them even though their patterns were close to the other exemplars. This finding mirrors the case where a child learns to label the category of dogs, but also learns to name the household pet (i.e. a particular dog, say 'Rover') to which he or she is frequently exposed.

This ability to represent both prototypical information about a category, and specific information associated with the exemplars of the category simulates the well-known familiarity and repetition effects found in studies of category learning (Reisberg, 1997, p. 185-193). The former refers to the fact that everything else being equal, familiar objects are easier to recognise than unfamiliar ones. In the context of McClelland and Rumelhart's (1985) study, familiar items are simply patterns that have been presented more frequently to the net and that are individually encoded in the weight matrix, so that the net will tend to respond stronger to these patterns than to those encountered less frequently.

Notice that the study serves to demonstrate how ANNs can be used to model category learning and generalisation in a distributed memory model. Several subsequent studies in psychology and neuroscience have relied on much the same network architecture (e.g. Small, Hart, Nguyen, & Gordon,
1995; Plunkett, 1995; Plaut & Shallice, 1993). For instance, the network in Small et al. (1995) made use of 77 semantic features to encode objects, in contrast the 16 features used in McClelland and Rumelhart's (1985) model. Likewise Plunkett and Shallice's (1993) model was similar in broad outline, but they made use of more complex network configurations and more complicated training regimes than McClelland and Rumelhart's network. Nevertheless, the research goal was much the same, that is, to show that a neural network adaptively acquires a distributed representation of concepts that mirrors aspects of human conceptual structure.

4.2.3 Broadbent's critique: Are ANNs just implementations of classical systems?

Although McClelland and Rumelhart's (1985) model simulates aspects of human memory performance, it has some limitations. The visual features presented as input to the model are based on arbitrary encodings. The modellers eschew the problem of attention, which is intimately connected to that of memory and learning. Also, the model is based on a simplification of the inductive problem because the authors assume that the input features are given whereas the real problem for category learning is to describe how an agent actually discovers the perceptual features associated with categories. Finally the model only deals with a relatively simple category structure containing a few members, so that the results of the study may not be generalisable in a straightforward way to the acquisition of complex categories with fuzzy boundaries and containing many members.

These issues are largely matters relating to the scale of the model, because it is mainly aimed at simulating the process of prototype abstraction, and not at yielding a detailed account of all the psychological processes attendant on category learning. A rather more serious critique of the model (than the points mentioned in the preceding paragraph) is presented in Broadbent (1985) who argues that in principle the network lacks theoretical value because it merely provides an implementation of Morton's (1969; 1979) logogen model. Adverting to Marr's (1980) description of levels of processing (which is discussed in Section 4.3), Broadbent (1985) points out that psychological models can be presented at two different levels, a level of computational description and a level at which the focus is on how the model is implemented in some physical system. To illustrate his position, Broadbent submits that a sharp distinction can be drawn between the way in which the visual system combines stereoscopic information, colour, and intensity gradients to compute an interpretation of the visual information, and a description of what happens below this computational level. He contends that different physiological theories might be developed to describe how this computation is instantiated in the visual cortex, but that these physiological theories are pitched at a different level than the more abstract computational theory, and should not be seen as competing with it. Broadbent's criticism can be condensed to two fundamental claims, which are set out below:
The McClelland and Rumelhart (1985) memory model does not really offer an alternative conception of memory processes to those implicit in Morton’s (1979) logogen model. In the original formulation a logogen was an internal recognition device for a word. Each logogen has a level of activity which increases as evidence is accumulated from contextual information for one of the senses associated with the word. When a threshold of activity is passed, the word is recognised in terms of the given sense, after which activation returns to a resting level. Words that are presented frequently will have a higher than normal residual level of activation, and they will thus be primed more easily on subsequent presentations of the word. The logogen model also accounts for repetition priming and the familiarity effect, but does so without relying on distributed coding or parallel processing.

According to Broadbent, the distributed model should be seen as a realisation of the logogen model at a level that approximates the physiological level. Unfortunately, according to Broadbent no detailed information about the physiological basis of memory is available, so that the model can only be evaluated as a fairly broad theory about the neural correlates of memory.

These points clearly have some merit because a cognitive model based on a parallel processing architecture is not necessarily more instructive or even more powerful from a psychological vantage point, than a classical computational model. Classical cognitive scientists such as Pylyshyn (1984) would argue that psychological theories are concerned with the virtual machine, the functional cognitive architecture underlying the brain’s parallel processing system, and that cognitive explanations can therefore be abstracted away from aspects of the neural system. The point is that connectionist theories tend to conflate the cognitive and neural system whereas Broadbent, along with some other researchers, hold that they should be kept apart for theoretical purposes.

In the light of such issues, Broadbent (1985) maintains that McCelland and Rumelhart’s (1985) model ‘merely’ implements Morton’s (1979) theory of logogens. According to him, it does not provide an independently formulated, and clearly articulated, cognitive theory. The points Broadbent raised in his paper, were subsequently set out in a more sophisticated format in a seminal paper by Fodor and Pylyshyn (1988), in which they contend that connectionist nets are basically associative systems which lack the power to cope with higher level cognitive functions such as language processing and reasoning. The application of connectionist modelling techniques to the realm of higher order cognitive functions will be considered in the next chapter.
4.3 THE IMPLEMENTATIONAL CRITIQUE IN PERSPECTIVE

Classical theorists draw heavily from an argument by Marr (1980) that an information processing system can be analyzed at different levels and that these levels are autonomous. From this they argue that the cognitive level is specifically concerned with algorithmic explanations and that such explanations are best formulated in terms of classical, symbolic approaches. The main issue to address is in what respect levels of analysis are separate, and what the significance of that (putative) separateness is. Such an analysis impacts on the connectionist approach in two major ways: firstly, it raises the question of whether ANN models are ultimately collapsible into more conventional (serial, von Neumann) approaches; secondly, it bears on what connectionist theories should look like-in, and in particular, whether they too are best formulated in terms of different levels. For some problems ANN modelling and classical approach complement each other. There are, however, major differences in basic assumptions that result in quite different theoretical emphases. A crucial difference lies in the somewhat different notion of explanatory level adopted in the two approaches. Marr (1980) characterized three levels of analysis associated with the explanation of cognitive functions as information processes, which has become very influential in cognitive science for thinking about computation. He delineates a:

- computational level of abstract problem analysis wherein the task is decomposed according to plausible engineering principles;
- level of the algorithm, specifying a formal procedure to perform the task, so that for a given input, the correct output results; and
- level of physical implementation, which is relevant to constructing a working device using a particular technology.

An important assumption underlying Marr's view is that a higher-level descriptions can be formulated without being overly concerned with details about the implementation (the neuronal architecture). For this reason, many classical theorists tend to view implementation as a lowly enterprise which does not bear directly on the development of theories of human cognition. On their view, the primary goal of cognitive science and psychology is to develop functional, symbolic accounts of cognitive processes, which capture the intentional nature of these processes, and implementation only comes afterwards, as an ancillary stage with much less theoretical importance. A critique of connectionism along these lines is set out in Fodor and Pylyshyn (1988), because they argue that the connectionist framework is not powerful enough to cope with some higher cognitive functions, and that the sole contribution of connectionist models to cognitive theory lies in offering a framework for implementing some parts of classical theories. The issue about implementation is thus to some extent emotionally charged, and carries with it a critique of the connectionist research programme by some classicists.
Two related issues emerge from this debate about levels of explanation. The first concerns the question whether ANNs can be regarded as autonomous cognitive models, and the other relates to the contribution that implementational theories make in cognitive science. For most classical researchers the greater neural plausibility of connectionist systems with regard to classical models is not at stake. They accept that ANNs embody interesting neural-like properties, but contend that these are not directly relevant to the goal of cognitive theory which, on their view, is to describe the characteristics of the cognitive virtual machine. They argue that this goal reduces to an explanation of the functional properties of the cognitive system, and such functional explanations can be presented without going into details about how the system is actually implemented in the brain (Fodor & Pylyshyn, 1988). Hence, according to the classical position even though connectionist systems may be more neurally plausible than classical systems, (this assertion needs to be qualified as we'll see in the next section), this has no direct bearing on their relevance to cognitive theory. Their main point is that ANN models do not yield an autonomous level of description, because as explanations of cognitive processes, they are reducible to classical models.

A point to note is that the reliance on level of explanation which classical researchers tend to highlight is not as straightforward as Broadbent (1985) would lead us to believe. Churchland & Sejnowski (1989, pp. 27-29) actually go so far as to question the viability of the notion of ‘level’ in the context of the brain sciences. Marr's levels and corresponding emphasis on the computational and algorithmic levels were born out of the early movements of artificial intelligence, cognitive psychology, and cognitive science, which were based on the idea that one could ignore the underlying biological mechanisms of cognition, focussing instead on identifying important computational or cognitive level properties. Churchland and Sejnowski (1989, p. 28) captiously refer to this doctrine of independence as “Marr's dream” contending that it is inappropriate for studying the nervous system. In the brain higher levels of analysis are very dependent on lower ones, because physical implementation details such as cytochemistry, morphology, and connectivity have significant effects on the choice of representations and algorithms that the system has selected.

The gist of the story is that there are different interpretations of the place that connectionist systems occupy in cognitive science. Classicists hold that connectionist models are in a sense just translations of symbolic theories into a format that takes the physical characteristics of the neural system into account. They claim that connectionist models do not make a contribution at the cognitive level of theorising, because classical models are expressive enough (as shown by the Church-Turing thesis) to yield a complete description of the computational aspects associated with human cognitive processes. ANN modellers, on the other hand, insist that their approach is aimed at describing psychological processes, but make use of a different (a non-classical) cognitive architecture (Horgan & Tienson, 1994).
4.3.1 In defense of connectionist modelling

Given the different perspectives on the relationship between classical and connectionist approaches, it is important to address the issue systematically. In the rest of this chapter, my main concern is to show that ANN models and classical systems rest on different assumptions about the mental representation of knowledge, and that the representational schemes underlying the approaches are sufficiently distinct for them to be regarded as yielding two different conceptions of cognitive architecture. The main issue is nicely stated in Chater and Oaksford (1990), and hinges on whether the computational and implementational levels of description are autonomous. One can argue that because ANN models are ultimately concerned with the implementation of cognitive processes in brain-like processing systems (i.e. parallel distributed mechanisms), the cognitive explanations emanating from connectionist research may not be trivially reducible to the serial processing, symbolic accounts associated with classical approaches. ANN models can only be collapsed into classical models if it can be shown that the explanations emanating from the use of connectionist systems are directly translatable into classical accounts. Only in the latter case can the classical level of explanation be regarded as autonomous.

In pure computer science, representations or data structures are often a given, determined by say the architecture of some abstract machine, or by the mathematical nature of the data that the algorithm operates upon. In cognitive science it is also often important to know that the algorithm can actually be carried out, and that it would yield a practical (as opposed to a purely theoretical) solution to a problem (more about this in the next chapter). Those parts of computer science where choices of representation arise, such as object-oriented programming, are also those with historically the closest connections to human cognition, via AI studies. In cognitive science, very little is really known about the machine and the ontology of the knowledge base is often obscure. The identification of an appropriate algorithm to simulate an aspect of human cognition, is often the best clue about the architecture of the hidden machine. The role that representational structures play in such architectures is underestimated by critics of the ANN modelling enterprise such as Broadbent (1985), and Fodor and Pylyshyn (1988). Some of the salient differences between connectionist and classical notions of memory 'representation' are considered in more detail in Section 4.4.

However, before dealing with representational issues, it is first pertinent to briefly consider two arguments that have been advanced to promote the connectionist approach. These arguments are both problematic, because even though they highlight desirable properties of ANNs, they cannot be used to build a convincing case for the distinction between ANNs and classical models in cognitive research. The first is the claim that ANNs are neurally plausible, and the second highlights their parallel processing capabilities.
4.3.1.1 ANNs are neurally plausible

ANNs mimic some of the features of biological networks, and much of their appeal comes from their neural plausibility. These models incorporate neural-like processing elements, are based on parallel processing mechanisms, and attempt to provide a characterisation of cognitive phenomena in terms of this framework. In fact, one of the attractions, or “allures” (see Section 1.2.2) of the connectionist paradigm is precisely that it may foster insight into the operation of brain-like computational devices (McClelland et al., 1986, p. 10). In contrast, classical theories abstract away completely from physical devices, so that the connection between the cognitive aspects addressed in these theories and the underlying neural substrata of the brain is obscure or, as Smolensky (1991b) puts it, a “total mystery”. At the very least then, connectionist theories fill an important niche in the cognitive science landscape.

The point is that the typical classical preoccupation with the algorithmic aspects associated with cognitive phenomena, at the expense of their implementational correlates, results in an incomplete explanation of psychological processes. Connectionist models are more closely tied to the brain’s physiology and could be used to express cognitive theories in a format that is broadly consistent with neural principles. One possibility is therefore to interpret connectionist nets as necessary adjuncts of classical explanations, without supplanting them. Nevertheless there are some difficulties with the neural plausibility argument:

☐ ANNs have some neural-like features, but also differ in significant respects from real neural systems. For example, in backpropagation networks the error values are propagated backward through a network, but there is no evidence that real neurons allow error signals to be fed back in this way. There is also no proof that the environment supplies the type of accurate training patterns required by supervised learning algorithms (Dawson, 1998);

☐ ANNs are often based on a large number of simplifying assumptions about the structure of neurons, about synaptic communication, and about network topologies, so that Douglas and Martin (1991) describe them as “stick and ball models” (cited in Dawson, 1998, p. 61). Still, there are some networks which are much more neurally faithful. Pulvermüller (1999) has shown that Hebbian networks can be used to simulate aspects of language processing, and the LEABRA system described by O’Reilly and Rudy (2001) is specifically designed to simulate a biologically realistic conception of aspects of human memory;

☐ Only a limited amount of progress has been made in understanding the brain’s information processing properties. Its sheer complexity has resisted satisfactory explanation, particularly in respect of some aspects of higher-order cognitive functions such as language and reasoning
Moreover, as Smolensky (1988) observes, the available data on brain functioning is typically not of the right kind. Most of the current neuropsychological data relate to the brain's structural properties and concern gross localisation of psychological functions. Much less is known about its functioning as a system, where much of the prevailing theorising is still relatively crude. Instead, cognitive science requires information about the brain's operation at a computational or systems level.

The point is that not enough is known about the cognitive-brain nexus to use neurological information to actually constrain cognitive theorising in a significant way. Hence because the contact between neuroscience information and cognitive-level explanations is still of a limited kind, it would be unwise to put too much weight in the neural-plausibility argument for ANN modelling.

### 4.3.1.2 ANNs are parallel processing systems

ANN modelling is often supported by arguments that processing in the brain is carried out in parallel, not serially as in classical, symbolic computational approaches. However, this is a muddle. Classical researchers assume that the cognitive system is divided into different modules and accept that processing in the different modules can occur in parallel. It is possible, although maybe not straightforward, for symbolic systems to process content in parallel by devolving processing to a variety of modules or submodules which operate independently of one another.

The main issue to address is what do we gain in explanatory power by postulating a parallel system. Cognitive explanations require that one peers inside a particular module to see how it operates, and algorithmic explanations in the classical style highlight the effective procedure, the mechanisms underlying cognitive processes. It has been shown that some supposed refinements to a Turing machine do not improve its computational capabilities. Making TM's non-deterministic, endowing them with multiple tapes, adding multidimensional attributes to the tape, or including multiple heads provably do not bring about a general improvement in the computational power of this system (Hopcroft & Ullman, 1979, p. 158-166). Hence the addition of a parallel processing mechanism does not necessarily render a connectionist model more powerful from an information processing perspective than a classical serial model. For this reason, Oliphant (1997) formulates the following lemma:

"...[conventional digital] computers on the one hand and connectionist networks on the other are exactly equivalent in processing capabilities. Whatever functions one of these kinds of machine can perform, so can the other"

As the discussion in the previous chapter has shown, any conventional digital computer endowed with an indefinite supply of memory, has the power of a Turing Machine. The ordinary desktop computer is the
most powerful conceivable kind of information processor, it can process any computable function if it is
given enough time to do so. ANNs are universal function approximators as shown in Chapter 2, and
consequently a network can be designed to process any (continuous) function, but even though ANNs are
powerful computational devices they are no more powerful than serial machines in terms of the type of
functions they can carry out.

4.4 KNOWLEDGE REPRESENTATION: LOCALIST VERSUS DISTRIBUTED
APPROACHES

A more promising line of attack is to consider the role of the representational schemes associated with
the two approaches in mode detail. Broadbent’s criticism downplays the difference between the
distributed representations employed in McClelland and Rumelhart’s (1985) model and the localist
encoding of information implied by the logogen theory of memory. There is at least some reason to
suppose that the distributional scheme is not compatible with the classical approach where specific
representations correlate with specific represented elements. Generally, in a symbolic representational
scheme the structure of the domain is mirrored in the constituent structure of the representation. Different
items are represented by different symbol structures. This is not the case with distributed representations.
The configuration of connection weights embody the knowledge encoded in the network, yet this
knowledge cannot be interpreted semantically in any straightforward manner. In fact, the net’s total
weight matrix simply yields the potential for generating specific, semantically interpretable activations.
For this reason, Haugeland (1991, p. 81) maintains that the superpositional scheme completely
undermines the notion of symbolic constituents associated with classical modelling, because “there is just
one big token representing many different contents at once”.

At the very least, the ANN and classical approaches yield two different metaphors for thinking about
conceptual structure, and these metaphors translate into different research practices.

4.4.1 Classical concepts of memory representation: The library filing metaphor

On the classical view, memory representations are faithful copies of the stimulus information that was
available on a past occasion, and are stored as discrete records, that can be retrieved separately, like a
book in a library or a page in a file cabinet. It is therefore based on the assumption that memory storage
involves a system of unique locations much like a library or filing cabinet in which each item has its own
unique storage location. In such a knowledge representational system, information is amenable to a
variety of well-understood symbol manipulation procedures and search techniques. Retrieving a memory
trace amounts to searching in the file cabinet to locate the appropriate representation, so that the memory
associated with original event can be re-experienced. The approach draws from what one may call a
‘pigeon-hole principle’ in terms of which information stored is located at a specific address. Searching for and locating the appropriate address at which a piece of information is stored is an important function in this approach, and specifically in computational models based on it. Because search processes play a primary role in the classical approach, Partridge (1995, p. 54) refers to it as the “symbolic search space hypothesis” of cognition.

The library filing conception of memory is a pervasive metaphor in cognitive research, but it suffers from some drawbacks. Because storage location is indexed by the name or ‘label’ associated with the item, it is difficult to explain how retrieval of an item based on partial information or cues can be performed with relative ease. Moreover, as the already mentioned study by Bartlett (1932) suggests, human memory does not function like a camera or a tape recorder because memories are often reconstructed and are not just direct copies of stimulus information. In classical models, the formation of a memory representation is assumed to reflect generalisation or abstraction from many experiences, but it is unclear how this abstraction process works so that general traces can be formed. The classical approach fails to provide satisfactory answers to such issues because it does not offer an explanation of how memory and learning are integrated in cognitive processes.

4.4.2 Memory processing in a distributed system

ANN models of memory are based on the ideas of soft constraints and distributed representations, and offer a somewhat different approach to memory and memory processes than the classical ideas of symbolic representations, and address-based storage. The notion of a distributed representational system associated with ANN models, comes in different guises. It may refer to a system where each unit is dedicated to a specific feature with no overlap between the representations. This scheme is distributed because the information associated with the concept is spread out over all the units representing its features, as exemplified by the ‘microfeature’ approach often thought to characterise PDP-type connectionist models (see e.g. the argumentation in Quinlan, 1991, pp. 220-222). A slightly different approach is a sparse distributed representational system in which each item is represented by a particular pattern of activation over a pool of neural units, and the state of the whole pool is relevant in determining what item is being represented. In the limiting case this type of sparse distributed system converges with localist schemes, as illustrated by Rumelhart and McClelland’s (1986) connectionist model of past tense learning. The primary representational items in this model are verb base forms and past tense forms. These are represented by sparse distributed activity patterns over pools of neural units, which represent features of verbs (Wickelfeatures) in a strictly local fashion.

The term ‘distributed’ is more commonly reserved for situations where multiple items are represented simultaneously over exactly the same units. In these superpositional schemes representations are
distributed and also coextensive so that it is impossible to partition the units into disjoint sets responsible for encoding distinct items. The most elaborate form of distribution is equipotentiality where a primary item is represented over a set of units and every part of that item is represented in superposed fashion over the same units, so that any subset of units represents the item in a holistic fashion. In an equipotential scheme, the entire representation can be recovered from any subset of the units representing it. The optical hologram illustrates this type of scheme. Holographic representational systems have been advanced by some researchers as a mechanism for simulating human memory (Metcalfe, 1993).

Some of the main representational schemes are illustrated in Figure 4.3 below. In the figure, C stands for category, and a1 to a4 denote attributes associated with a category.

![Diagram of five different representational schemes](image)

**Figure 4.3** Five different representational schemes.

**Key to interpreting the five schemes**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td><strong>Symbolic localist</strong></td>
<td>One unit represents the concept</td>
</tr>
<tr>
<td>b</td>
<td><strong>Distributed, but quasi-localist</strong></td>
<td>Each unit represents an attribute of the concept</td>
</tr>
<tr>
<td>c</td>
<td><strong>Sparse distributed</strong></td>
<td>The representation of an attribute is distributed over two units</td>
</tr>
<tr>
<td>d</td>
<td><strong>Superpositional</strong></td>
<td>The representation is superimposed on a small number of units</td>
</tr>
<tr>
<td>e</td>
<td><strong>Equipotential</strong></td>
<td>Each unit represents the whole concept</td>
</tr>
</tbody>
</table>

The superpositional version of distributed representations was first elaborated in mathematical models.
of memory (see Murdock, 1992), and now forms the basis of many of the popular distributed representational models of cognitive processes (see Rumelhart et al., 1986a; Smolensky, 1988). In this scheme, the representations associated with different items are spread out over exactly the same units, and a whole set of units takes part in the representation of an item, with this representation manifested as a pattern of activity across units. Moreover, in this superpositional type of representation, the pattern of the activity of an individual unit does not correspond to any particular feature of a concept — at least not traditionally conceived features such as 'has four legs' in relation to the category dog but may be implicated in the representation of different features as illustrated by (d) in Figure 4.3.

4.4.3 Coarse coding

In some respects the distributed style of representation seems to be inconsistent with the way in which localist representations store information. For example, one of the interesting aspects of some distributed representational structures is that they exhibit a property known as coarse coding (see Hinton, McClelland and Rumelhart, 1986, pp. 91-94). The basic idea underlying coarse coding is that individual units are sensitive to many different aspects of the input, and input features activate many different units, so that a representation is inherently entangled, resulting from a coalition between units.

The fact that a particular unit is active is not very informative in this type of system, because the presence of the input can only be inferred with confidence if a high percentage of the units become active. Note that in some distributed schemes (e.g. example b in Figure 4.1) each unit signals the absence or presence of a specific feature, so that coarse coding is not necessarily associated with distributed representational structures. However, coarse coding is a property of the superpositional representational scheme used in PDP nets. One can think of the units in networks employing a coarse coded representational system as receptors or basins of attraction that are sensitive to the receptive fields of a number of different input units (Bechtel & Abrahanson, 1991, p. 54). Still, the concept of coarse coding is not very intuitive, because the idea that a single unit may be implicated in the representations of different, non-overlapping inputs is difficult to conceptualise.

Coarse coding appears to be a feature of some neural structures and is inter alia exploited by the brain's visual representational system (Hinton et al. 1986). The eye makes use of three types of cones to register colours. The cones are sensitive to different wavelengths; one type responding maximally to wavelengths corresponding to red, another to green, and the third to blue. Although each type of cone responds maximally to a single wavelength, it responds less vigorously to a range of longer and shorter wavelengths. Indeed, the response of the three kinds of cones overlap considerably, and it is this overlap which allows us to see so many colours. We see maroon, orange, magenta, and so forth as a result of the mixture of the outputs from the three types of cones. Coarse coding is this method of exploiting the
degree of overlap in responses from units (or entire networks) with different sensitivities, to generate outputs with precise values. As the visual system's colour representation scheme illustrates, coarse coding is a very efficient way of specifying information because only three kinds of cones are needed to specify a range of colours. It is also intimately tied to the idea of constraint satisfaction which is characteristic of connectionist information processing. Each unit imposes a weak constraint, and a response is only generated when multiple weak constraints are added together and exceed the given threshold.

The real advantage of coarse-coded, distributed representational systems over localist styles of representation has to do with representational capacity. Localist representations impose a rather rigid framework on the conceptual contents of a model, because it is associated with the idea of a fixed and discrete inventory of concepts. In contrast, a coarse-tuned, distributed system is more flexible, and has some useful properties.

☐ It is relatively robust to damage (i.e. it provides for 'graceful degradation' in the face of damage) because several units are involved in the representation, so that damage to any particular unit is rarely fatal. Clearly a certain amount of resilience to damage is an important property for human memory and cognition, because it ensures that some level of performance may remain intact after injury to the system.

☐ Distributed representation is consistent with what is known about the neural basis of cognition. There does not seem to be much support for the proverbial 'grandmother cell' theory of memory trace, implied by some localist models, among brain scientists (Edelman & Tononi, 2000). The distributed representations used in ANNs therefore have at least some compatibility with the brain's neural representational system.

☐ Distributed, coarse-coded systems yield a powerful and economic representational system, because the number of patterns that can be represented grows exponentially with the number of neurons (i.e. representational units). For example, if we assume that neurons are binary (e.g. they can have values of 1 or 0), there are $2^n$ possible patterns for $n$ neurons in a distributed system.

☐ Perhaps the most important property of distributed representation is that although the activity of individual neurons is ambiguous, the pattern of activity across neurons need not be. That is, the pattern created by a set of highly redundant, coarsely tuned units can have a greater resolution than that which can be achieved by the same number of more finely tuned, less redundant, units. Distributed systems achieve such high resolution because various units make small, but fine-grained contributions to the overall pattern.
However, distributed representations also have some disadvantages. Several critics have pointed out that parallel and distributed theories of memory are often unable to explain certain phenomena under conditions of sequential learning (Lightfoot, 1998). The learning algorithms associated with these theories are generally constructed to find a set of values that can represent items that are shown many times to the networks in mixed order, and at intervals distributed across the set of items shown. However, when all the presentations of each item are massed, when learning is sequential, learning of later items will disrupt memory for earlier items. Human memory experiments predict that some kind of retroactive interference is likely to occur under typical learning conditions, but ANNs seem to be far more gravely affected by interference and exhibit what McCloskey and Cohen (1989) term “catastrophic interference”. McCloskey and Cohen (1989) demonstrated such interference in a simulation of an experiment by Barnes and Underwood (1959) in which they trained a multilayer, feedforward net first on one list of stimulus-response pairs, the AB set, and then subsequently on another set of pairs, the AC set. They found that unlike humans who can retain some of the associations in the AB set when they learn the new set, the network rapidly lost all ability to retrieve the AB items after only a few trials on the AC items.

Ratcliff (1990) observed similar catastrophic interference in an ANN model of recognition memory. The model also tended to track the information presented last in the list, and exhibited rapid forgetting of the well-learned items that had been presented earlier. Although the observation of such catastrophic interference is problematic for connectionist approaches to memory, it is not disastrous, because various solutions have been proposed (e.g. as discussed in Thrun & Pratt, 1998). For instance, researchers have found that such interference can be avoided by adjusting the connection weights to reduce the overlap between competing responses (Sloman & Rumelhart, 1992). Also, Carpenter (1997) has developed a mathematical theory which suggests that an appropriate choice of learning laws for connection weights can markedly reduce interference without modifying the architecture of the model.

4.5 A SUBSYMBOLIC LEVEL OF REPRESENTATION

The argumentation in the previous section suggests that distributed memory systems give rise to a conception of memory and a way of thinking about memory processes that differ in some salient respects from the address-based notion associated with the classical approach. In a distributed system, memory traces are not stored in any particular location, but are diffusely encoded over many units. Rather than imagining that particular units encode information, the idea is that memory is stored in the relationships among such units. The operation of a distributed memory system can be understood by analogy with the way in which a filter extracts individual frequency components from a complex acoustic waveform. Even though the individual frequency components are completely intertwined with one another, the filter is able to detect the presence of the specific component it is tuned for. Likewise, as long as the individual memory traces are sufficiently different from each other, a distributed system can operate as an effective
storage and retrieval system. The interesting thing about such a system is that when memories are not completely independent, they interact with one another so that storing one memory may affect another. Information that is related to, but different from previous information, tends to evoke the previous pattern of activity. Thus unlike conventional symbolic memories, these systems have similarity and the ability to generalise as central attributes. The role of similarity is discussed in more detail further down.

If it is granted that ANN models embody a new conception of the processes used to carry out memory operations, then it evidently does not invoke entities such as symbols and rules, but concepts such as activation flow, weight strengths and distributed representations. One can therefore argue that these systems should be viewed as pitched at a unique level of representational description, distinct from the classical notion of symbolic operations. For example, the case study of the IAC network shows that ANNs instantiate a rather unique computational mechanism. With multiple representations all stored in the set of connection weights, retrieval can no longer be regarded as a process of searching for a single representation and retrieving it. Instead retrieval is a reconstructive process that seeks to satisfy multiple constraints simultaneously. Neisser (1967) likened this type of memory to the work of a paleontologist who reconstructs a dinosaur from some fragments of bones (i.e. data) and a body of pre-existing knowledge about dinosaurs (i.e. theory). Similarly what is retrieved from memory is constrained by the cues that trigger a recollection and by specific details of the to-be-remembered event or object itself, but also by related general knowledge, memory traces of similar events and so forth. These ideas are broadly consistent with PDP type representational systems, because in such networks many stored representations necessarily interact during memory processing, yielding a type of reconstructive memory (as illustrated by the IAC case study).

4.5.1 Probing below the symbolic level

Because of these differences between symbolic and distributed ANNs (i.e. PDP models), Rumelhart, McClelland et al. (1986) argue that the connectionist approach helps to elucidate the microstructure of thought, suggesting that ANNs can be used to explore the operation of massively parallel and distributed mechanisms, and ‘fine-grained’ cognitive processes. In similar vein, Hofstadter’s (1985, p. 632) contends that perceptual processes take place below the 100ms interval, and that these are ignored by some classicists such as Simon whose work focus almost exclusively on cognitive phenomena such as planning, and problem solving, occurring above this interval. Hofstadter (1995) further argues that perception is a central aspect of cognition, that analogy is a key component of perception, and consequently that approaches in AI and cognitive science based on the assumption of fixed symbolic structures are overlooking substantial ‘pre-processing’ involving perceptual and analogical processes performed by the cognitive system. He developed a computational model (called “COPYCAT”) in which a number of microprocessing modules, called codelets carry out most of the cognitive tasks. In the computational
architecture codelets are weighted by urgency, but precisely which of the many active codelets is run next is determined randomly. Once the weights have been suitably adjusted, the model exhibits the desired behaviour as an emergent property of the interaction between individual codelets. To ensure that the approach has a chance of scaling up from micro-domains to ‘real’ cognition, it is important to make sure there are no brute force searches, because a combinatorial explosion of possibilities may render such an approach infeasible; randomness is used so that not all paths, but only the currently most promising paths, are searched. The degree of fluidity - how much the program is willing to explore weird paths, is controlled by a temperature parameter: the hotter the temperature, the more random the search. But unlike the simulated annealing technique used in the Boltzmann machine, the temperature is set by the program itself, by how ‘happy’ it is with the partial solutions it has built so far. The happier it gets, the more it concentrates on its current choices. Hofstadter (1995, p. 34) calls this search technique the “parallel terraced scan”.

Smolensky (1988) draws a similar distinction between finer and broader levels of cognitive operation, and advocates what he called a “proper treatment of connectionism”, embracing mainly the PDP-type approach to ANN modeling. He maintains that ANNs fitted with distributed representations should be viewed as instantiating a subsymbolic approach, which is directed at clarifying operations that take place lower than the symbolic level of description typically adopted in classical models. The distinction between the symbolic and subsymbolic is somewhat problematic, because the latter term is not clearly operationalised in Smolensky (1988). Nevertheless, the main ideas can be characterised roughly as follows: In the symbolic paradigm descriptions used in the representations of situations are built of entities that are symbols both in the semantic sense (they refer to categories of external objects) and in the syntactic sense (they are operated on by the manipulation of such symbols). In the subsymbolic paradigm, such descriptions are built of subsymbols, which are fine-grained entities such as the units and weights in an ANN model. In a symbolic approach, symbols constitute the base objects of semantic interpretation. The symbols are thus representations, so that the fundamental objects that the system computes with are alleged to contain the meaning for the system as well. If a system as a whole was currently representing dog, then each individual representation would stand for say, a particular feature of a dog. There would be a representation for 4 legs, for a cold, wet nose, for a tail, and so forth. Each of these representations would directly refer to, and symbolise a feature of the dog. In the subsymbolic approach, the processing occurs at a level lower than the symbolic one, so that semantic content only emerges from this subsymbolic level. As Chalmers (1992, p. 34) explains it, the computations take place at one level and in some manner blend together to give rise to meaning at a higher level:

In a symbolic system, the objects of computation are also the objects of semantic interpretation. In a subsymbolic system, the objects of computation are more fine-grained than the objects of semantic interpretation.
In the latter system, representations are distributed over a set of units, without there being a one to one relation between individual units and the meaning of symbols that emerge from them. The entire network, or at least a significant portion of it, is the representation. Furthermore, depending upon the network's pattern of activation, it can be any of a number of representations. This type of representation is therefore non-atomic and distributed. If a subsymbolic computational system were to represent a concept such as *dog*, each unit would only contribute to the overall representation, reflected in the entire system's particular state of activation, and only this pattern of activation would correspond to the concept. Altering any single unit in a symbolic systems results in a representational change. However, in the case of subsymbolic representations, even though each individual unit plays a causal role in the distributed representation, none of them carry the crucial semantic burden. All that they do is “crunch and gurgle” the information in a particular way so that the system as a whole can have meaning (Chalmers, 1992, p. 35).

The main theoretical rationale underlying the ‘symbolic’ and ‘subsymbolic’ distinction is that symbolic descriptions are too rigid and hard. Hofstadter (1995) and Smolensky (1988, pp. 1-5) both suggest that cognitive mechanisms can be sufficiently flexible to model cognition only if they exploit soft constraints that emerge from the operation of a large number of fine-grained processes at a subsymbolic level. On the view that ANNs are associated with a subsymbolic level of description, there is no strict isomorphism between classical and connectionist models because the latter work with a different conception of semantics. Classical theories assume that each individual representation has semantic content, whereas the units in an ANN only exhibit semantic content at a higher level than the unit. In the case of such nets, semantics is an emergent property associated with activity patterns of collection of units and it does not reside in the symbolic content of individual representations as such. Because conceptual level explanations are assumed to be only approximate descriptions of the activity patterns which occur at a subconceptual level, Smolensky (1988; 1991) argues that connectionist architectures are properly understood as refinements rather than implementations of classical architectures. It is pertinent to note that because the subsymbolic level is offered as a new cognitive level of theorising, it does not just reduce to the neural level. Smolensky’s (1988) explicitly advances subsymbolic processes as cognitive in content, but lower than the typical symbolic level of description. The conception of the relationship between the neutral, symbolic and subconceptual levels of description suggested in Smolensky (1988) is depicted in Figure 4.4 below:
Figure 4.4 An interpretation of the relationship between the neural, subsymbolic and symbolic levels.

Smolensky’s main argument is that connectionist systems are not simply implementations, because complete formal and precise descriptions of the cognitive processor are generally tractable at the subconceptual rather than the conceptual level. Conceptual level theories tend to be incomplete in that some processing details are omitted, or imprecise in that they yield only an approximate or idealised description of what are in fact massively parallel, statistical computations that taking place at a lower level.

The subsymbolic paradigm appears to be a radical departure from the classical conception of representation, but it is not necessarily in conflict with the symbolic level of theorising. An important postulate associated with the notion of a subsymbolic level is that symbolic descriptions or explanations of cognitive phenomena may actually be built up out of many smaller constituent. Entities that are typically represented in the symbolic approach using symbols may derive from a large number of subsymbols, but for cognitive descriptions it is often important to analyse subsymbolic models at a higher level, to “amalgamate” the subsymbols into symbols (Smolensky, 1988, p. 3). However there are some difficulties with the idea that the two approaches (i.e. symbolic and subsymbolic) can be reconciled in this way. One problem is that it is not clear at all how a cluster of subsymbols could form a traditional symbol. Symbols have arbitrary labels, such as a string of letters which are atomic, discrete, and static. In contrast, a symbol in the subsymbolic paradigm is distributed over a collection of subsymbols, each associated with a continuous numerical value. How can a symbolic representation ever emerge from such
non-symbolic processes?

Typical symbols in a symbolic model might be the letter string \textit{waiter} or \textit{customer}. These symbols may be placed into structured relationships with other symbols, and may be bound to a variety of values during the course of processing, but the forms of the symbols themselves never change. The symbolic model would work equally well if \textit{waiter} were replaced by \textit{xyz} throughout the model's data structures, because \textit{waiter} is simply an atomic label possessing no internal structure of its own. In contrast, a typical symbol in a subsymbolic model might be the pattern of continuous values \([+0,3343\ -0,1277\ +0,7654\ -0,5435\ +0,5655]\). During processing, this might evolve to a slightly different but similar pattern \([-0,3383\ -0,1377\ +0,7554\ -0,5535\ +0,5631]\), which behaves in a closely related way to the original symbol. However, the subsymbolic model would produce very different results if this symbol were replaced by some other arbitrary pattern. Hence, even if it is true as Smolensky suggests that the symbolic and subsymbolic levels are related, the nature of the relationship clearly needs clarification.

\textbf{4.5.2 The role of similarity}

It is generally accepted by cognitive researchers that similarity and categorisation are important aspects of human cognition. The process of classifying objects is a fundamental feature of most human pursuits, and the idea that we classify together those things that we find similar is both intuitive and popular across a wide range of disciplines. Similarity-based models of classification abound in cognitive psychology as accounts of human performance and in AI and machine learning as the basis of practical applications. However, despite their centrality, the notions of 'similarity' and 'categorisation' are still comparatively poorly understood. The major cause for this has been that until recently many disciplines have tended to study these only as adjuncts to larger questions and without making contact with research in other fields. Even in cognitive psychology, where these phenomena have been tackled directly, much discussion of similarity and categorisation has been fragmented across a variety of specialist domains, with related aspects such as conceptual categorisation, metaphor and analogy, decision making, problem solving and memory being studied largely in isolation (see Hahn & Chater, 1998).

Similarity is perhaps the main representational principle driving the distributed scheme associated with ANNs, because in the course of training, a multilayered ANN's hidden layer begins to exhibit a similarity gradient. For example, in multilayered ANNs trained with a gradient descent algorithm such as backpropagation, each hidden unit constitutes one axis of a vector, and each activation pattern across the hidden units is a set of points in an \(n\)-dimensional vector space, where \(n\) is the number of hidden units. In learning to classify patterns during the training process, the network partitions the weight space into discernable sub-regions (or attractors in the case of recurrent networks) corresponding to its categories of classification such as \textit{dog} or \textit{cat}. Similar items of information interact in such a way as to reinforce
aspects that are common, and to cancel aspects on which they differ. This means that weight space configurations cluster similar things, and also that weight configurations may be sensitive to very fine differences between objects, consequently dividing the activation space between similar objects. An activation space will also be a similarity space in as much as similar vectors will define adjacent regions in space. Similarity between objects represented in such a system can be reflected by similarity in their representations, which entails proximity of positions in activation space. Similarity in representations is therefore not an accidental feature of ANNs, but an intrinsic property of these systems, and one which is exploited when cognitive phenomena such as memory processes are modelled. As Churchland (1995) points out, networks function in very high dimensional spaces, since weight space will have as many dimensions as there are weights, and activation space for hidden units will have as many dimensions as there are hidden units. A consequence of these design properties of ANN systems is that even though only a finite training set is used, the network may yield good answers to inputs it has never seen before. Provided the input information has some resemblance to previously seen inputs, the network will be able to categorise it appropriately because vectors which are in close proximity in the weight space will be dealt with similarly, allowing the network to cancel small differences and to make generalisations. The ability of ANNs to process information about stimuli in terms of similarity makes them suitable for modelling aspects such as face recognition, which is difficult to cope with using conventional symbolic approaches (see Cheng, O'Toole & Abdi, 2001; Young & Burton, 1999).

In general the similarity metric is a good rule of thumb, and lacking other information, it is reasonable that similar things should be dealt with in the same way. In any case, similarity is usually the most useful metric to use for constructing categories and for evaluating membership to categories. On the other hand, there are clearly cases where two patterns which resemble each other superficially should be treated differently. A child learning to read should be able to discover that although how and row are spelled the same, they are pronounced differently, and that despite surface similarities one must learn to treat some words differently. The problem is that although similarity is often a good starting point for making generalisations, there are many circumstances where surface similarities may lead one astray, and a more abstract and functional kind of similarity is often needed. Moreover, there are many philosophical properties associated with judgments of similarity. As Goodman (1955) has pointed out, any two objects are alike in a virtually infinite number of ways. For example, a fly and a skyscraper are both concrete objects, are less than a kilometre in length, are approximately 200 000 000 kilometres from the sun, and so forth. Determining similarity in terms of common properties is therefore problematic unless plausible restrictions are imposed about the properties that are relevant. Nevertheless judgments of similarity is a common facet of everyday cognition, and the fact that ANNs can use pattern recognition processes to classify entities based on similarity, makes them appealing as models of cognition.
4.5.3 Cross-talk and analogy in distributed memory systems

In memory processes a phenomenon called cross-talk occurs when an attempt to activate a particular trace results in another trace, which is in some respects similar to the intended one, being activated. The phenomenon is endemic to large distributed systems in which large numbers of similar traces are stored. The resulting errors tend to resemble the memory slips characterising everyday memory, and the sensitivity to context typically exhibited by human memory processes (Schacter, 2001). The interesting thing about cross-talk in distributed systems is that it manifests a crude (i.e. very elementary) form of analogical reasoning. In such systems the activation of a unit may sometimes activate other units with which it is only weakly connected. If the criteria for similarity is weakened in this way a response is often simply incorrect, but in some cases the response may have an analogical connection to the intended one because analogy is, after all, only similarity in a weaker sense.

The essence of analogical thinking is the re-interpretation of one entity or event in terms of another. This requires a system which is able to pick out similarities on different dimensions (such as form and function). The topic of analogical reasoning has traditionally been dealt with in classical approaches. However, lately there has been considerable interest in using ANN models to simulate the processes associated with analogical reasoning. In a connectionist account of analogy, the extent to which two traces share sub-patterns of activity will have some effect on the probability that one may trigger activation of the other - just as in cross-talk. If two traces are seen as overlapping on some dimension, one will be reminiscent of the other pattern, thereby creating the possibility of analogical linkage. Of course, this is a rather simplistic view of analogy, and clearly a more complex theory is needed to account for all the facets of analogical reasoning. However, the point is that the simple explanation ‘flows’ naturally from connectionist information processing mechanisms, and indicates how overlap in activation pattern may underpin seemingly related phenomena which are typically dealt with separately under classical explanations (see Eliaimith & Thagard, 2001).

One of the most influential theories of analogical reasoning was presented by Hummel and Holyoak (1997) who proposed a connectionist model called LISA (Learning and Inference with Schemas and Analogies). A key assumption incorporated into this the model is that there is an important distinction between analogical access and mapping. Access involves a retrieval process based on competition among various stored analogues in permanent memory, of which only a few of become accessible to consciousness during any specific retrieval attempt. In contrast, mapping entails comparing the features of a single stored analogue against the current target in working memory. A distinguishing feature of LISA is that the initial processes involved in mapping take place without the intervention of conscious strategies, in a relatively automatic fashion, invoking the use of implicit rather than explicit memory
4.5.4 Integrating memory and learning

Because similarity and generalisation are properties of distributed memory systems, the distinction between memory and learning becomes blurred in these systems. In classical, symbolic models of cognitive processes, it makes sense to distinguish between memory as a store of learnt information, and processing operations in terms of which information is used for memory operations, and new memories are added to the existing representations. The processing aspect may be closely related to the storage aspects of memory, but the two aspects are nevertheless conceptually distinct. In ANN models of memory, the storage and processing aspects become integrated, so that there is no clear distinction between them, because the learning and adaptive capabilities of ANNs are essential aspects of the connectionist research programme. The ‘memory’ of an ANN model is best viewed as a continuously updated system in which new information are written over older information so that it is difficult to talk about a stable, fixed structure as in classical models. In ANNs, memory is essentially a process and storage and learning aspects occur in tandem. For this reason, Mandelblit & Zachar (1998) have suggested that conceptual units should not be construed in terms of the rather static entities associated with classical approaches, but with more flexible, dynamic units, whose interpretation is mainly defined in terms of their interactions with context and other co-occurring elements (i.e. they have a ‘fluid’ rather than static semantics). The assumption about dynamic properties of conceptual units is interesting, but it is fairly obvious that concepts and words must have at least some core semantics otherwise communication would be impossible. Meanings cannot just be dependent on context, but must have at least some stable aspects as well.

To account for the relatively fixed aspects of meaning, connectionist researchers have typically appealed to prototype theory as suggested by Churchland’s (1995), and the McClelland and Rumelhart (1985) simulation which modelled a prototype approach to representation. It is useful to consider the rationale for prototype theories of meaning briefly, as it has proven to be a fertile area for exploring connections between connectionist models, and cognitive linguistic approaches to conceptual structure (Collier, 1998).

Classical, symbolic approaches to conceptual structure are based on the tenet that there are natural kinds of entities and that each kind is defined by shared essential properties (see Lakoff, 1987). Members of a category are therefore bounded by similar attributes. Some cognitive linguists and psychologists have argued that a prototype conception of meaning is more in line with human conceptual structures. On such a view, membership to a category is not determined by abstract shared properties but by similarity to a prototype. Rosch (1975, 1978) showed that naturally occurring categories such as FURNITURE, ANIMALS, have such a prototype structure. The idea of prototype structure has been further elaborated
by cognitive linguists such as Lakoff (1987) and Langacker (1987) who argue that many linguistic phenomena (e.g. phonological properties, sentence structure, schemata) can be analysed as instances of categorisation around a prototypes. In the latter case, there are no necessary and sufficient conditions for membership which are shared across the members, instead, membership is determined by virtue of relationship to the ideal case. Churchland (1995) maintains that this ability to naturally represent prototypes is one of the most appealing aspects of ANNs as tools for modelling cognition, because learning algorithms such as backpropagation have the effect of changing the weights so that the network finds a point in weight space where errors are minimised. This may lead to the building of prototype representations, because the system is most sensitive to information falling about the central tendency of similar inputs. One can think of these regions as prototypes, ‘hot spots’ with fuzzy boundaries. Any prototypical point represents an extended family of relevant (but not individually necessary) features that collectively unite the relevant class of stimuli into a single kind.

The ability of ANNs to cope naturally with prototype structure makes them compatible with the ideas of cognitive linguists such as Lakoff (1987) who has argued strongly against what he calls the “objectivist metaphysics” underlying classical theories of meaning. He maintains that the typical classical idea of fixed, symbolic meanings is unsatisfactory, and that the standard model-theoretic account of meaning invoked by classical researchers (see Cann, 1993) may lead to a proliferation of unwanted models. The situation arises under the assumption that the predicates of the language can be assigned wild interpretations which are ‘unintended’, and this runs contrary to our normal intuition about the relative stability of meanings. Even worse, the assumption that meanings are relatively fixed would appear to be a standard assumption adopted under the classical approach, so that the paradox of unwanted models would appear to undermine the classical notion of representations. Lakoff’s argument against the ‘objectivist’ position is presented in Appendix C. Lakoff (1987) further contends that metaphor in language is best construed deriving from a fixed pattern of conceptual correspondences across conceptual domains. As such each mapping defines an open-ended class of potential correspondences across inference domains. When activated, a mapping may apply to a novel source domain and characterise a corresponding target domain. A metaphor system imposes a massive number of constraints on a conceptual system, and these limit the possibilities of how situations can be conceptualised and for how novel linguistic utterances are understood. These findings are at odds with symbolic traditions in AI and classical cognitive science. In these fields it is assumed that thought is purely a matter of fixed symbol manipulation, but metaphor has an image-schematic basis which characterise constraints on the generation of novel metaphor. These constraint-satisfaction aspects of metaphorical mappings are more compatible with connectionist approaches, than with symbolic models. For example, Thomas and Mareschal (1996) developed a simulation of simple “A is B” metaphors using a three-layered, feedforward network. Their work is only programmatic, but nevertheless shows that the distributed mechanisms of ANN systems can be used to explore aspects of metaphor creation.
4.6 PUTTING IT ALL TOGETHER

The chapter started by discussing claims that ANN models are best viewed as theories of implementation, showing how higher level functional explanations may be realised in brain-like mechanisms. Constrained in this way, ANN explanations would be fully compatible with higher-level classical accounts, instead of being alternatives. However, if one consider some of the properties of connectionist models, then viewing them as equivalent to classical may be somewhat simplistic. For example, the representational structures that have been proposed for human memory in classical approaches often presuppose many of the same properties as computer memory, and thus constrain the type of possible cognitive architectures. For example, the assumption that memory and attention involves search (which features inter alia in the ‘zoom lens’ theory of attention - Sternberg, 1999, p. 81) as an access mechanism relies on a unique storage principle. The conceptualisation of human memory in the framework of distributed ANN networks is rather different from the classical perspective, because the coarse coded, superposition storage principles associated with these networks are not consistent with the idea of a single trace for every memory event. Instead different memory traces are represented by distributed patterns over the same processing units. Moreover the processing of memory also occurs as activations distributed over many units. Clearly the ANN approach yields a different metaphor and vocabulary for thinking about memory and learning.

There are some drawbacks distributed representations because the complex interactions between processing units in such systems often makes them rather opaque to the researcher. It is also difficult to identify and interpret the internal representations that ANNs develop through learning, and even more difficult to investigate precisely their fundamental properties and possible limitations. Nevertheless, distributed representations offer functionalities similar to classical symbols, but achieved in a very different way. Like symbols a distributed representation can be identified by its own form (an activation pattern) and each form typically correlates with a distinct object in the semantic domain. Contrary to classical symbols, distributed representations are not totally arbitrary because the distributed scheme is driven by similarity, so that similar objects will have similar representations in the net’s weight space. Moreover, distributed mechanisms generalise well, they are resistant to damage and incomplete, noisy or ambiguous stimuli, when they are damaged or presented with extremely poor stimuli their performance degrades gradually rather than catastrophically. All of these features can be linked back to the essentially constructive nature of distributed representations. As distributed representations are constructed from patterns of much simpler characteristics representations of similar stimuli, they will share many of those characteristics. Novel stimuli will therefore evoke responses similar to those whose existing representation they share features with (i.e. the system will generalise). In a similar manner, a stimulus which is novel because some of its characteristics are unknown or corrupted will also evoke a response similar to other stimuli sharing most in common with it (i.e. it is resistant to damage). Finally, if much of the stimulus
is missing or parts of the network have been damaged the information remaining can still be used (i.e. it exhibits a graceful degradation of function). Given these properties of distributed systems, it seems that there is reason to doubt the claim by Broadbent (1985) and other classical researchers that ANNs are just implementations and of no real consequence in cognitive theorising.

However, so far I have only considered rather low-level connectionist approaches to memory and conceptual structure. For connectionism to be viewed as an alternative to the classical theory, the basic low-level assumptions about networks (as distributed systems relying on equations defined over vectors and weight matrices) have to be exploited to construct high-level theories that make novel claims about information representation and processing. One of the main attractions of connectionism is that networks are capable of performing interesting computations without invoking explicitly coded symbolic representations or structure-sensitive rules for processing. The lack of symbolic representations is critical to some of their more alluring properties such graceful degradation, lack of brittleness, the ability to exploit soft constraints and to make generalisations. However, as the next chapter will show, the lack of specific facilities for dealing with symbols is a drawback when ANNs are used to model aspects of cognition such as language and reasoning, where such symbolic entities appear to play a crucial role.
CONNECTIONISM AND HIGH-ORDER COGNITIVE FUNCTIONS: MODELLING LANGUAGE

The previous chapter focussed on ANN models of cognitive structure, and presented some reasons why connectionist modelling should not be viewed as a purely implementational endeavour in which the main focus is on porting classical, symbolic level theories to parallel processing, brain-like architectures. In contrast, a conception of connectionism, deriving mainly from Smolensky (1988; 1991a), was presented which attempts to formulate a coherent framework for ANN modelling based on the notion of a subsymbolic level.

However, most of the psychological phenomena considered in the chapter related to perceptual, memory, and categorisation functions, which are often thought to reflect lower-order psychological processes. They are assumed to be more primitive processing systems than those associated with the so-called higher cognitive tasks such as language interpretation and reasoning (see Quinlan, 1991, p. 195). Some of the assumptions underlying this division between higher and lower order cognitive processes are somewhat problematic. Quinlan traces the notion of higher and lower-order functions back to the levels-of-processing idea articulated in works such as Hyde and Jenkins (1973), who argued that semantic processing (e.g. interpreting the meaning of a word or sentence) involve a deeper-type of analysis than visual recognition processes (e.g. scanning the letters in the word). This notion of deeper levels of processing was inspired by early theories of generative grammar, such as Chomsky’s (1965) conception of syntactic structures, and some crucial aspects of the initial generative conception and its theoretical descendants in semantics and memory have since been abandoned (see Chomsky, 1981; Pinker, 1994). Despite the fact that it may lack theoretical force, the idea that reasoning processes, language understanding, and problem solving are complex because they require some sort of conscious mediation, and are thus of a higher order than processes, such as perception, which unfold relatively automatically, has some intuitive appeal. It therefore may serve an heuristic purpose even in the contemporary cognitive science context.

Classical theories cover the whole spectrum of the cognitive field, but the approach has achieved notable successes in cognitive domains associated with the higher-order functions such as problem solving (Newell & Simon, 1972), reasoning (Johnson-Laird & Byrne, 1991) and language (Chomsky, 1993; Pinker, 1994). These are also the processes that have proven recalcitrant under earlier theoretical approaches such as behaviourism (see e.g. Chomsky, 1959), and that, as will be shown later in this chapter, pose the greatest challenge to connectionist models in the cognitive sciences.
In this chapter I first present a survey of the field and then move on to consider some specific issues that have arisen in the context of ANN models of the structural elements of language. These are in a sense some of the more difficult obstacles that ANN researchers have to face in order to show the feasibility of the connectionist paradigm in cognitive science.

5.1 CONNECTIONISM AND LANGUAGE: GENERAL ISSUES

ANN modelling of language processing is to say the least highly controversial. Some researchers have argued that language processing can be understood in connectionist terms, but several proponents of the classical approach have questioned the viability of the ANN paradigm. They maintain that it does not foster an understanding of human language processing mechanisms. For example, Chomsky recently dismissed the ANN modelling paradigm as a framework for linguistic theorising, asserting that its contribution to our understanding of language "is about zero" (cited in Smith, 1999, p. 135). The controversy is particularly heated because ANN researchers have advanced their networks not just as an additional technique for studying or modelling language processes, but as an alternative to symbolic systems for developing theories of language. Perhaps the most significant reason why some classicists maintain that ANNs are ill-suited to modelling language is based on the argument that these systems exploit associative mechanisms, and are therefore incapable of adequately representing structural aspects such as the constituent structure of sentences. Of course ANN researchers dispute this claim and point out that their systems are not just old-fashioned associative mechanisms, but exploit powerful nonlinear capabilities (e.g. Elman et al., 1996, p. 47). This debate has become especially significant because it highlights the concerns that some researchers have about the feasibility of using neural network systems to model higher-order cognitive functions more generally.

Symbolic models of language are strongly tied to logic as a formalism. The reliance on logic is not only evidenced by research in core areas such as syntax and semantics (Barwise & Perry, 1983; Cann, 1993; Gazdar, Klein, Pullum & Sag, 1985; Pollard & Sag, 1983; Montague, 1974), but also in more applied fields such as the development of natural language processing interfaces and discourse understanding systems (Allen, 1995; Cohen, Morgan & Pollack, 1990; Gazdar & Mellish, 1989) where logic is the formalism of choice. In fact, until recently the journal Computational Linguistics - which is perhaps the primary journal serving the natural language processing research community - concentrated almost exclusively on articles invoking some type of logic formalism, typically of a more advanced variety than first-order predicate logic. Allen (1995) in a well-known textbook surveying natural language processing research, does not even mention ANNs as a technology for developing language understanding systems. In the recently completed survey on human language technologies, the contribution of connectionism to language modelling is acknowledged, but is assumed to be mostly restricted to speech processing technologies rather than semantic or syntactic processing (Uzkovert,
1996). Some researchers working in computational linguistics therefore appear to harbour the general sentiment that classical, symbolic computational models offer the best route for making progress both in modelling human language abilities, and on the related technological task of developing intelligent natural language processing interfaces.

To anyone at all familiar with neural network research, the low profile that connectionist systems have in natural language processing applications may seem surprising. Language has always been a popular theme in connectionist research as already manifested in the PDP books (e.g. Rumelhart & McClelland, 1986), and attested by subsequent research covering various aspects of language use (Cree, McRae & McNorgam, 1999; Plunkett & Marchman, 1993; Plaut, McClelland, Seidenberg & Patterson, 1996; Seidenberg & McClelland, 1989). One of connectionism's main forte is in speech technologies, such as the NETtalk system developed by Sejnowski and Rosenberg (1986), and the highly publicised past-tense model of Rumelhart and McClelland (1986), which illustrated how a domain-general ANN learning algorithm can be used to extract language-related rules automatically from relevant data.

There are also several collections of articles available which demonstrate that ANNs can be used to model a variety of aspects associated with language processing (e.g. Reilly & Sharkey, 1992).

5.1.1 Some early ANN models of language processing

In cognitive science studies of aspects of cognition such as memory where some biological constraints are available, research has focussed mostly on modelling human behaviour. However, in the case of language research there is much less useful data from neuroscience available and much of the attention has focussed rather on modelling and understanding how language operates as a system. Data is taken from linguistics and psychology rather than from neuroscience. In the latter case, ANNs must compete head-on with classical, symbolic models of language processing. Moreover, because classical, computational approaches to language processing have a long tradition, researchers working on language in linguistics, cognitive psychology, and AI have proceeded from the viewpoint that symbolic models set the goalposts for connectionism, but not the reverse (see Levelt, 1990). On the other hand, many researchers involved in ANN applications of language processing have a more radical agenda; to challenge and supplant, rather than to simply reimplement the symbolic approach. There is also some evidence that although symbolic computational models and the formalisms associated with them provide a tried and tested platform for doing research in language, the symbolic approach has reached a plateau in some areas. This may be one of the reasons why many researchers in the natural language processing field are now experimenting with statistical approaches (see Manning & Schütze, 2000, pp. 15-17). For this reason, there has been some shift in interest to ANN models, because they implement a form of statistical computation, and thus provide an expanded toolkit for language researchers (Miikkulainen & Dyer, 1991).
Many aspects of language exhibit quasi regularities, regularities which naturally hold but admit of exceptions. In a symbolic framework, quasi-regularities can be captured by symbolic rules associated with explicit lists detailing the exceptions. Symbolic models often incorporate this distinction with the aid of separate mechanisms for regular and exceptional cases. In contrast, ANNs may provide a single mechanism which can learn general rules as well as their exceptions as the Seidenberg and McClelland (1989) model, which is described below, shows. Also, because ANNs learn from exposure to a domain they naturally complement corpus-based approaches in which natural language processing systems are applied to large portions of texts to discover patterns in the data (Charniak, 1993). There has lately been some success in developing text-based ANN systems, as exemplified by an ANN technique called Latent Semantic Analysis, presented in Landauer and Dumais (1997) and Kintsch (2001), which has been used for making lexical comparisons between texts.

It is worth pointing out that one of the dominant influences in mainstream linguistics is Chomsky’s generative linguistics which is associated with the assumption that language ability is innate. Chomsky is vigorously opposed to any form of empiricism (see Chomsky, 1980) and argues that language competence unfolds under the constraints of a genetic programme. His position is somewhat complex because he acknowledges that children must acquire the lexical or vocabulary component. He nevertheless maintains that structural aspects are not learned by mere exposure to language, but that humans have preknowledge (in the form of a genetic programme) of the structural component governing language. For this reason Chomsky (1980) has argued that mere exposure to language is not sufficient to induce the grammar of a natural language because the stimulus is too impoverished to direct the learning process. In similar vein, most generative linguists assume that at least some aspects of syntactic knowledge is innately determined, and that language is not learned totally from scratch. It would be naive to reject the nativist thesis completely, but the learning capabilities inherent in many ANN systems make it possible to explore some of the assumptions underlying the generative position. This has even resulted in interesting collaboration between generative linguists and connectionist researchers (Prince & Smolensky, 1997).

However even if there has been some contact between connectionist and classical positions, there have also been considerable debates and controversies between proponents of these approaches. As will become evident a little later in this chapter, there are significant problems in applying ANN models to some aspects of language processing, and this places a damper on the enthusiasm with which these models have been adopted by the natural language processing community.

5.1.2 Modelling language acquisition

One of the great allures of ANNs is that they can learn. This makes such models useful for modelling
aspects of human cognitive development. Plunkett, Sinha, Muller and Strandsby (1992) trained an ANN to represent a visual image with a label using the architecture displayed in the figure below.

![Network Diagram](image)

**Figure 5.1** The network used by Plunket et al. (1992) to model vocabulary acquisition

As the diagram shows, the model is an auto-associator because it has the same number of inputs as outputs with a condensed internal representation. The condensed representation in the hidden layer is used to force the network to abstract salient information in the input data, filtering out the noise in the data as it were. Conceptually the approach here is similar to factor analysis in that the network is required to discover the underlying latent structure of the input information presented to it.

At various stages during training the network was tested for comprehension and production. Production corresponded to it generating a label when an image was presented to it, and comprehension entailed generating an image when it was given the input corresponding to a label. Plunkett et al (1992) found that the network’s comprehension preceded its production just as is the case with children. They also found that the network’s learning occurred in a non-linear way. Initially learning was slow, with just a steady increase in both production and comprehension measures of vocabulary. However, with further exposure to the training data a sudden spurt occurred. A similar spurt is typically found with young children, and is usually attributed to the maturation of the language system in the brain (Newport, 1990). However, the fact that the network also produced such a vocabulary spurt even though it has no language system, suggest that it may be an artefact of the learning process itself.

Plunkett and Sinha (1992, pp. 227-228) present the following explanation. During the early stages of
training, the network is still attempting to make sense of the problem domain to which it is exposed, and discovers solutions for isolated aspects of the general learning problem. However, as the number of isolated solutions increases it begins to find out about the structure of the problem domain, and discovers the natural clustering of the objects. For example, the network begins to treat objects that share many attributes as belonging to a similar category. Once a critical-mass of image-label associations had been discovered leading to a preliminary categorisation of the input, the learning process suddenly becomes easier and a spurt is obtained.

5.1.3 The question about rules: Seidenberg and McClelland's (1989) study

Many aspects of language can be described using rules. This is particularly evident in morphology and phonology. Adding the suffix ed to a verb as in shout + ed often changes it from the present tense to the past tense. Unfortunately, there are exceptions (irregularities) to almost every rule, (e.g. eat is transformed to ate, fight changes to fought, and go becomes went in the past tense). Linguists and psychologists have tried to account for this observation by postulating a dual route principle: one route is based in a system of rules, the other includes a set of explicit exceptions. For this reason, traditional two-route models of reading aloud postulate two independent processing mechanisms which are assumed to be implicated in the conversion of orthographic input to phonological output that takes place during the reading process (Coltheart, Curtis, Atkins, & Haller, 1993). The assumption that two independent mechanisms are involved stems from two simple observations:

☐ People are capable of pronouncing letter strings they have never seen before such as slunky or avelin. This suggests that all skilled readers are equipped with a mechanism consisting of general rules of pronunciation which is used to generate a plausible reading of the pertinent string; and

☐ Skilled readers have no difficulty in finding the appropriate pronunciation for words, such as pint or handkerchief which do not obey these general rules. This suggests that there is a stock of specific knowledge which is applied to the pronunciation of individual words.

A typical assumption is that these two methods, the rule-based and the lexical-based methods, involve two separate pathways. The lexical route uses a store of information about specific words to generate a pronunciation via semantics, and the non-lexical route invokes a set of pronunciation rules. The general reasoning underlying the postulation of two different routes is that non words such as kint do not have an entry in the lexicon so that it can only be read aloud with the aid of pronunciation rules. Some evidence from neuropsychology support this interpretation. There is a striking difference between the patterns of reading errors made by two groups of patients who have become dyslexic as a result of brain
injury (see Coltheart et al., 1993). One group, called phonological dyslexics, reads words without difficulty, but cannot produce a pronunciation for non-words. A second group called surface dyslexics pronounces regular words and non-words correctly but make errors on irregular words, tending to regularise them. The contrast is easily explained by the two-route model. It is assumed that phonological dyslexics can still use the lexical route but have lost the rule-based route. In contrast, surface dyslexics have an intact rule-based route, but have lost access to the lexical route.

Connectionist models formulated within the context of Smolensky's (1988) subsymbolic approach challenge these assumptions. A distributed ANN does not contain anything akin to a lexicon, where specific information about particular inputs is stored, separate from other information. Instead information in the net is distributed across the weight matrix, and is superimposed over information associated with other inputs. To explore the assumptions underlying the two-route model Seidenberg and McClelland (1989) developed a two-layered feedforward network to simulate the reading process. In their study, reading is construed as a mapping from one domain (orthographic information) to another domain (phonological output), without the incorporation of a distinct lexicon. The network was trained to pronounce all the monosyllabic words in English with the aid of backpropagation.

Once trained on this vocabulary, the model gave an impressive quantitative fit to many empirical findings about factors influencing the speed at which skilled readers pronounce different words. For instance, it correctly predicts the effects that the consistency of the spelling to sound pattern and the number of different words with similar spelling pattern have on the time it takes to pronounce an individual word. The crucial point is that the network learned the reading task without being taught any specific rules, and without incorporating a special mechanism for dealing with the exceptions.

Building on the Seidenberg and McClelland (1989) model, Plaut, McClelland, Seidenberg, and Patterson (1996) developed a series of simulations to further investigate the orthographic-to-phonological decoding process associated with reading aloud. They constructed a network containing 105 input nodes, 61 output nodes, and 100 hidden units, and used a training set consisting of almost 3000 monosyllabic words. The words comprised regular as well as rule-exception words. Just like the Seidenberg and McClelland (1989) model, their network succeeded in learning the input-output mapping without the inclusion of a separate mechanism for dealing with exceptions, which again appears to refute the postulate of a dual-route mechanism. However, Plaut and his colleagues concede that the network may have divided itself into two sub-networks during learning, one for processing the regular words and the other for dealing with the exceptions. To test for this possibility they cut away or 'lesioned' individual hidden units and measured the effect of the lesioning on the error in the pronunciation of each word. They found that the units were responsible for the errors associated with the standard and exceptional cases to much the same degree. From this they concluded that both types of words were processed by the same mechanism.
5.1.4 Simulating dyslexia with attractor models

Some researchers have applied connectionist techniques to the simulation of the neuropsychological processes associated with reading deficits. In this type of modelling exercise, a net is first developed to simulate a normal process such as the orthographic-phonemic mapping and semantic decoding which occurs during reading. After this the neural net is lesioned (i.e. some units and/or links in the net are destroyed) by the researcher in order to explore the pattern of deficits that the lesioning brings about in the model. As will be shown presently, these connectionist simulations offer explanations for patterns of cognitive deficits which have been difficult to explain using classical style cognitive models. Moreover, the connectionist models make quantitative predictions about the performance deficit resulting from the lesioning, while alternative, non-connectionist approaches can only describe these deficits in relatively vague qualitative terms.

In the condition known as deep dyslexia, a patient produces a word which is semantically related to the target word, but has no phonological similarity to the correct response. When shown the word 'dog', a patient might respond ‘Spaniel’. Evidently the stimulus word is processed because the response is semantically related, but there is no phonological relationship between the target word and the actual response. The traditional interpretation for this type of deficit is that there are separate semantic and non-semantic routes from orthography to phonology, and that the patient has lost the non-semantic processing route. However, deep dyslexics also produce visual errors by, for instance, reading 'insect' as “insists”. (see e.g. Marshall & Newcombe, 1981). In this case the patient appears to have relatively intact visual decoding skills, but has lost semantic information. So the traditional interpretation would be that the patient has lost the semantic processing route and kept the non-semantic one. To add to the confusion, deep dyslexics also produce errors such as responding “skirt” to 'shirt' which suggests that they have kept partial information about both the visual and semantic characteristics of the target word.

The co-occurrence of such seemingly independent error types presents difficulties for traditional box-and-arrow models of cognitive information processing. Hinton and Shallice (1991) created a connectionist simulation of these processes. The model contained a layer of 68 semantic units (which they called "sememe units"), a layer of 28 grapheme units, a layer of 40 hidden units and a layer of 60 "clean up" units, with recurrent connections to the layer of semantic units. The architecture of the model is shown in the figure below.
Figure 5.2 A simplified representation of Hinton and Shallice’s (1991) attractor net.

The model was trained using a set of 40 words, 8 from each of 5 semantic classes. Each word was defined by a positive value on a subset of the semantic features. The task put to the model was to activate the correct set of semantic features (15 on average) for the object described by each input word. The net was first taught to produce correct orthography to semantics mapping by backpropagation, and was then lesioned by removing some units and connections, as well as randomly changing the value of some weights.

The recurrent connections were added to introduce an attractor structure into the model. The concept of a dynamical system and the relevance of the associated mathematics for describing connectionist models will be taken up in Chapter 6, so that for now a very brief definition suffices. In the connectionist context, an attractor can be conceptualised as a state, a set of output unit activities to which a net will gradually move if its is allowed to evolve during successive processing strategies. A simple feedforward net does not have attractor states because there is only one computational pass through the system during each learning sequence so that there is no opportunity for the state of the output units to evolve. However, a recurrent network can change its state as activity cycles around in the system. In feedforward nets each unit only computes its state (activity level) once in processing an input as the activation passes from the input to the output layers. In recurrent nets units provide feedback to earlier layers so that units in the net have to update their state repeatedly, because changing the state of a unit may change the input to earlier units. As a result the pattern of activity over the entire network changes over time in response to a fixed input. The network settles gradually into the appropriate final pattern of activity that corresponds to the appropriate interpretation of the input. This process can be conceptualised as movement in a multi-dimensional state space in which each dimension encodes the state of a particular unit. At any instant the current pattern of activity over all the units in the network is represented as a particular point in this space, and the net’s evolution over time corresponds to a
movement in state space from an initial to a final point for each pattern of activity. Different input patterns lead to different final points or attractors in state space. In such an attractor network there is a region in state space around each attractor (its basin of attraction). When the activity of the net falls anywhere in this region, it will move to the attractor during settling.

The Hinton and Shallice (1991) network was trained on a set of 40 words, and during training developed a set of attractors relating the 40 initial semantic states produced by the orthography of the words to the 40 semantic states representing the meaning of the words. Weight changes which create attractors turn out to have other consequences for the structure of semantic space. For instance, they lead to the creation of attractor basins. An activation pattern can be viewed as a trajectory through weight space, so that the trajectory of the activation entering a basin will be pulled by the attractor to the heart of the basin. The result of a lesion to a model is to change the shape of the attractor basin. Hinton and Shallice’s (1991) model showed that the co-occurrence of visual, semantic and mixed errors is not mysterious at all, but an inevitable consequence of lesioning a connectionist net that uses attractors to map orthography to semantics. Hinton and Shallice found that the lesioned model made several types of error which emulate that of the deep dyslexic patients:

- visual, when the words are related in terms of visual rather than semantic features (e.g. cat \rightarrow cot);
- semantic, when the retrieved word is semantically rather than visually related to the target (e.g. cat \rightarrow dog);
- mixed, when the retrieved word has both visual and semantic similarities to the target (cat \rightarrow rat);
- other, when the response is unrelated to the target (e.g. cat \rightarrow mug).

The visual, semantic, and mixed errors were all produced at a rate greater than chance, and this held for all the lesions performed on the model. There was a tendency for the proportion of errors to change as a function of the lesion site, and there were proportionally more visual errors with early damage and proportionally more semantic errors with late damage. However, this pattern corresponds to observed differences across patients in the neuropsychological literature on dyslexia (see e.g. Coltheart, Patterson, & Marshall, 1987).

5.2 FROM LEXICAL PROCESSING TO SYNTAX: THE SYSTEMATICITY DEBATE

One of the basic assumptions underlying the physical symbol system hypothesis is that symbols can be combined to form more complex structures (see Section 3.3.1). This type of compositional structure is exemplified by human, or ‘natural’ language, because natural language exhibits a combinatorial
syntax and semantics. What this means is that words can be combined into more complex structures such as collocations, expressions or sentences, and the meaning of the words contained in such structures determine their meanings.

Most of the problems associated with the application of ANNs to deal with the syntactic, or structural properties of language stem from a general difficulty ANNs have in dealing with variables. In order to cope with syntax, one should be able to form a general category for parts of speech and be able to determine whether an individual word can fill that position. This ability entails classifying some aspects of a sentence (e.g. subject of sentence) as a variable with slots that can be filled by many individual tokens of the right kind. For example, speakers of English know that *man, horse, and rugby player* can all fill a slot for ‘animate subject’ in a sentence such as *He ran across the field*. The simplest explanation of why standard ANN systems have trouble learning and implementing variables is because they do not have separate tokens for variables and their values. Hence, unless they are specifically designed they cannot distinguish between variables and values. Input enters the system as activations on input nodes, it gets transformed by weights and activation functions, and finally emerges again as activation on output units. The whole system operates simply by modifying and passing activations from input to output, propagating values. Standard ANNs crunch values, not variables.

Fodor and Pylyshyn's now often cited paper (1988) highlighted this difficulty that standard ANNs (i.e. ordinary multilayer, feedforward networks trained with backpropagation) face. They identify a feature of human intelligence which they call “systematicity”, and contend that it poses serious difficulties for connectionist approaches to language. Fodor and Pylyshyn offer very little by way of a general characterisation of the concept systematicity, but mention several examples of putatively systematically related cognitive capacities. They point out, for example, that anyone with a command of English who understands a sentence such as *John loves Mary* would also automatically understand the related sentence *Mary loves John*. From the classical point of view, the connection between these two abilities can easily be explained by assuming that when the sentences are parsed, the constituents ('John', 'loves' and 'Mary') are identified and their individual semantics and roles in the sentence are determined. Because the related sentence *Mary loves John* has exactly the same logical structure as the first one (the same constituents are involved, but their roles are simply reversed) it follows that it will also be parsable by anyone capable of understanding the first one. They then argue in detail that connectionist approaches do not account for systematicity. Although connectionist models can be trained to be systematic, they can also be trained, for example, to recognize *John loves Mary* without being able to recognize *Mary loves John*. Since connectionism does not guarantee systematicity, it does not explain why systematicity is found so pervasively in human cognition. Fodor and Pylyshyn (1988) conclude that systematicity may exist in connectionist architectures, but where it exists, it is no more than a lucky accident. The classical solution is much better, because in classical models systematicity is pervasive and “comes for free”.

134
The charge that connectionist nets cannot explain systematicity is initially quite plausible. However, the view has been criticized lately by several researchers (e.g. Aizawa, 1992; Chalmers, 1993). One point common to these rebuttals is that symbolic processing models have exactly the same feature which was supposed to deny connectionists an ability to explain systematicity, for there are also classical models that can be programmed to accept *John loves Mary* and reject *Mary loves John*.

In Fodor and McLaughlin (1995, p. 113-114) the challenge to ANNs is put in this way: “to explain the existence of systematic relations among cognitive capacities without assuming that cognitive capacities are causally sensitive to the constituent structure of mental processes”. They see this challenge as implying a dilemma, because if ANN modellers fail to account for systematicity, it would imply that the connectionist paradigm does not provide an adequate theoretical basis for modelling language processes, because - on Fodor and McLaughlin’s (1995) view - systematicity is an integral aspect of language. On the other hand, if connectionist researchers do develop models which account for systematicity, they would have to include processes that are sensitive to constituent structure of mental representations in their networks. But in so doing they would end up constructing an implementation of the underlying classical model (which of course has knowledge of constituent structure as a dedicated, built-in part of the cognitive architecture). In the latter case, the connectionist account would simply be a parallel processing version of the serial, symbolic theory, without adding any new theoretical content in the modelling process.

### 5.3 Specialised ANNs for Modelling Sentence Processing

According to some proponents of the classical, symbolic paradigm such as Fodor and Pylyshyn (1988), the property that truly distinguishes ANNs from symbolic models relates to the way in which representations are manipulated and transformed. In classical, symbolic systems representations are governed by rules that are themselves represented symbolically. These rules may take the form of production system instructions such as ‘If A then B’, or they could be formulated as expressions in a logical system. The rules in a symbolic system operate on the representations in a manner that is sensitive to their constituent structure. Symbolic representations can have a complex structure in that they are composed of symbols that stand in syntactic relations to other symbols. Rules can operate on a symbolic representation by virtue of its structure alone.

In ANN models, the relation between representations and rules is very different. Representations are not construed as formal objects, and are consequently not subject to direct manipulation. The rules in an ANN are inherent in the equations that direct the network’s behaviour, and they operate at the level of individual processing units. These ‘rules’ specify how units compute their activation, how activation is transmitted to other units, and how the weight coefficients are modified. Individual processing units
interact with and causally affect other units, and intelligent behaviour emerges from the way in which the interacting units are connected, it does not arise from the direct interaction and transformation of representations. It is this distinction between ANNs and symbolic models that gives rise to the controversy over the need for explicit rules and symbolic representations in models of natural language processing as implied by Fodor and Pylyshyn (1988) in their critique of connectionism.

Since the publication of Fodor and Pylyshyn's (1988) paper, the study of language has been viewed as a crucial test of the adequacy of ANN models for cognitive theory. Traditionally language is regarded as the quintessential case of a rule-governed, combinatorial system (see Chomsky, 1980), because it is composed out of a finite number of discrete elements (phonemes, morphemes, words). These elements are combined systematically to create larger hierarchical structures (e.g. sentences) and the rules that combine to create these larger structures are sensitive to the syntax of their constituent elements. In such sentences, there are rules that specify the agreement between nouns and verbs, that regulate the tense of the expressions and that control the semantics of the compositional process by which the sentence is formed.

The challenge put to ANN researchers is to implement complex structural relations such as constituency in a system that lacks explicit rules specifying how structures are to be combined. As the models described will show, some progress has been achieved, but ANNs are still far from having convincing techniques for sentence processing on the table.

5.3.1 St. John and McClelland's model

St John and McClelland (1990) constructed a connectionist model that learns to assign semantic representation to English sentences presented as input. The model is trained by backpropagation to extract the situation described by each sentence, mapping the syntax onto a canonical, logical form associated with notions such as agent, patient and action. The assumption is that the relationships between these notions capture a semantic level representation of a sentence.

The actual sentences given as input were somewhat simplified structures in that articles were omitted and only singular nouns were included. Certain prepositional phrases were included, and these input sentences are fed into the network in pre-segmented constituents. As each constituent is processed an inspection is made via a series of probes on an internal set of units in order to establish whether the net has extracted the relevant representation of the target situation. Because it is usually not possible to predict the entire target representation on the basis of isolated constituents, the network is forced to learn associations between individual constituents and particular objects or relations in the target situation.
The input layer consists of a set of vocabulary units and a set of word position units. Each sentence constituent is represented in the input layer by activating one of the vocabulary units, corresponding to a part of a sentence such as a noun, verb and so forth. Vocabulary units consist of items such as nouns, verbs, prepositions, adverbs, and ambiguous words - i.e. words, which can function as a noun or verb, depending on the context. The word position units were used to indicate a constituent's position relative to the verb. The inclusion of the latter units is very controversial, but St John and McClelland remark that they have obtained comparable results without relying on such position indicators. Sentences are presented to the network by activating pairs of units in the input layer, each pair corresponding to a word and a word position for a given sentence constituent. The network's output in response to a given constituent consists of a set of values representing respectively thematic role, concept (or word meaning) voice, and some subset of possible semantic features. There were nine thematic roles, including agent, action, patient, and location. As a well-trained network processes a sequence of constituents corresponding to an entire sentence, an appropriate corresponding sequence of representations for concepts, thematic roles and semantic features will appear in the output layer.

The network design includes several hidden layers, of which two are of special interest. The first of these is called the sentence gestalt (SG) layer, which is trained to produce a distributed meaning representation for each of the sentences in the training corpus. The second is called the context layer, and retains a copy of the most recent content of the SG layer. The contents of the SG changes as each new constituent is fed into the input layer. During each iteration of the training algorithm, the contents of the input layer together with the most recent contents of the SG are propagated upwards to the SG layer. The resulting new contents of the SG are then propagated upwards but are also copied back to the context layer (awaiting the next iteration). The context layer enables the network to retain accumulated information about previous states of the network, and this accounts in part for the context sensitive behaviour eventually displayed by the fully trained net. Training the network entails repeatedly probing the SG layer as each constituent is processed to establish what series of values it produces on the output layer. Backpropagation is employed after each probe in response to the errors revealed through probing - and works on the entire set of thematic-role/concept pairs pertinent to the meaning of the sentence. Thus an input sentence such as The boy throws a rock with his hand is associated with a thematic role representation such as [(agent/boy), (action/throw), (patient/rock), (instrument/hand)], and this representation is required at the time the sentence is processed.

In general, the training presupposes that the learning system possesses concepts applicable to all objects and relations in the target situation, that the appropriate thematic roles, as dictated by the situation in question, can be assigned to these concepts. The assumption is that the learning system is able to realise that the same concept can play different roles, depending on the situation. Hence, the system is assumed to possess a systematic, compositional method of conceptualising the situation associated with a sentence before training begins. As a result, St John and McClelland's model cannot explain how
such a compositional scheme develops. The training corpus included sentences containing prepositional phrases, but during the testing phase simpler sentences lacking prepositional phrases were used. Two experiments were conducted to test for systematicity of behaviour— one syntactic and one semantic. The test used to probe the syntactic aspect associated with systematicity involved only 10 objects and 10 reversible actions, but 2000 sentences (both active and passive forms) were used for training. Given the small sample of words and the large training set used during the ANN learning process, it is reasonable to assume that during training the net experienced every word in almost every possible position, so that its generalisation abilities appear to have been limited.

5.3.2 Elman’s simple recurrent net

Elman (1990) describes a study in which he used a simple recurrent network to simulate the learning of some structural aspects of language. Simple recurrent nets are able to overcome some deficiencies associated with the standard multilayered nets trained with backpropagation which cannot cope with temporal relations. These recurrent nets are now generally assigned an important place in connectionist approaches to language and time-related aspects of cognition. In a survey of articles published from 1990 to 1994 Elman’s paper was the most widely cited paper in psycholinguistics, and the 11th most cited paper in psychology (research index database - http://www.bigat.com). The simple recurrent net is particularly important because it has been used to motivate an argument that the problem of variable binding may not seriously jeopardise ANN approaches to cognition because nets with recurrent connections can perform variable binding. An example of a simple (also called a “partial” or “Elman”) recurrent net is shown in Figure 5.3.

![Diagram of a simple recurrent ANN]

**Figure 5.3** A partial recurrent ANN as proposed by Elman (1990)
Elman's network was given the task of predicting the next word in the string of words given as input. The input sequences were produced by a symbol-manipulating algorithm, which would presumably have to be replaced by a connectionist system at a later stage. At each time step, the model was presented with a given word in a sentence and the target to predict is the next word in that sentence. The network was not trained on all possible continuations of sentences simultaneously, but on a specific continuation of a particular sentence at any given training step. The net contained an input layer, a hidden layer, a context layer, and an output layer as shown in the simplified representation in the figure above.

Some but not all versions of the model also included two additional transducer layers, but the simplified version captures the essential aspects of the model and thus suffices for purposes of this exposition. The model functions as follows: Given input at time \( t \), the hidden unit activity is fed forward to the output units but also back to the context units. At time \( t + 1 \) the input to the hidden units is the new input plus the previous activity pattern to the hidden units, which is cycled back from the context units. At the conclusion of each time step the contents of the hidden layer are copied (using fixed rather than trained connections) to the context layer and this content provides the model access to temporal information.

Input words are encoded locally as a vector composed of 0s and a single 1 corresponding to the unit activated by the word. The network is at any given time frame trained on a particular set of input words, and the context units are used to encode the temporal, or sequential, relations in the input set. The context units therefore serve as a memory which enables the network to learn temporal relations among the inputs.

Elman created a stream of consonants and vowels. The consonants \( b, d, \) and \( g \) were arranged in a random sequence of 1000 letters. Each consonant was replaced so that \( b \) became \( ba \), \( d \) became \( dii \), and \( g \) became \( guuu \). Thus the sequence \( dbgg \) would become \( diibaguuuguuu \). The resulting sequence had structure in that although the position of the sequence was random, the letters following each occurrence were fixed. The research goals was to discover whether a network could discover this structure so that it could generate the next member of the series as output? Each letter was presented to the network as a six-bit vector. The simulated network had six input units, twenty hidden units, and six output units. For a given letter presented as input, the expected or target output was the next letter in the sequence.

During a training phase, the input letter sequence was presented 200 times, and weight changes were calculated using backpropagation. The ability of the network was then assessed with a new letter sequence. Experimentation verified that the network could learn the appropriate substitution showing that it is sensitive to the structure in the input arrays.

Given that a simple recurrent network could detect regularities in strings of letters, it is reasonable to ask whether it can learn something useful. A next simulation investigated this possibility. A training set was created in the following manner: from a population of 15 words, 200 sentences were generated and
concatenated to form a string of 4963 sentences. Each letter was represented on the input units as a five-bit vector. A network of five input units, twenty hidden and twenty context units, and five output units was created. Training followed the same regime in which the goal was to establish whether the net could learn to predict the next letter in the sequence following a given letter as input. The pattern of errors was striking: high at the beginning of a word, but decreasing with each subsequent letter in that word. This shows that the network has learned something about the orthographic regularities found in words, based on its exposure to the co-occurrence statistics between letters in the words in the training set.

This net has been applied to a variety of temporal learning tasks. For example, Elman trained the network on a semi-realistic artificial grammar that included a variety of dependencies such as subject-verb agreement (The boys play vs the boy plays). Elman argued that such dependencies between the subject and verb are particularly important because they figure prominently in earlier arguments against statistical models lacking explicit grammatical rules advanced by Miller and Chomsky (1963). See also the discussion in Section 3.1.3.1 in this regard. Once the model was trained it was capable of predicting plausible continuations to strings such as dogs run and even more complicated sequences such as boys who run after dogs.

There are, however, some difficulties in evaluating the model’s performance. The primary quantitative measure of the net’s performance presented by Elman was to compare the continuations predicted by the model with those actually occurring in the test corpus. Unfortunately the test corpus was generated in the same way as the training corpus so that it is not clear to what extent the test corpus actually probes the model’s generalising ability as opposed to its ability to memorise. There is also no formal analogue of how well the model performs compared to a human on the given task. Nevertheless, Elman’s work shows how temporal sequences can be learned and reproduced by recurrent networks and it provides one solution to the problem of how long term representation of serial order may be incorporated in ANN models.

5.3.3 Pollack’s recursive auto-associative network

Pollack developed an auto-associative network which exhibits a degree of systematicity in a limited domain. The architecture of Pollack’s recursive auto-associative memory network (RAAM), is essentially that of a simple auto-associative memory. The network is an auto-associative model with equal sized input and output layers, but with a smaller hidden layer so that the net develops after learning a condensed, distributed representation of the input. Pollack showed that a RAAM network is capable of abstracting nested structures such as parse trees on the output layer. A parse tree captures the syntactic structure of sentences in a canonical format. For instance the nested list
( ( Det (Adj Noun)) (Verb (Prep (Det Noun)))))

generates a variety of different sentences such as:

_The hungry dog chases after a rabbit_
_The young baby plays with her dummy_

A RAAM net is recursively trained, in stages, to auto-associate a bit form representation of such parse trees with themselves. The training entails first training the net to auto-associate the simplest constituents, and it is then exposed to progressively more complex constituents. In the final stage the net is trained on the entire parse tree. Thus, in the case of the example above, the net is first exposed to the constituents (Adj Noun) and (Det Noun), then to more complex structures (Det (Adj Noun), (Prep (Det Noun), (Verb (Prep (Det Noun))) and finally on the whole expression.

Pollack ran an experiment in which a RAAM net was trained to autoassociate 13 ternary tree structures encoding 13 English sentences of increasing complexity. Ternary trees are triads of the form \((\alpha \beta \lambda)\) where \(\alpha, \beta\) and \(\lambda\) represent either atomic symbols, or another triad of the same form. In Pollack’s application \(\alpha\) was however always atomic because he encoded relational terms in the first position. An example of such a ternary tree, which includes an embedded structure, is shown below:

(Thought Pat (knew John (Loved Mary John)))

The 13 structures comprising the input set ranged in complexity from:

(\(\operatorname{Love}\) Pat Mary)

to

(\(\operatorname{Saw}\) (Is (\(\operatorname{Mod}\) Man Short) (Thought Man (Saw Man John))) Pat)

Using a representational scheme where each terminal symbol in the parse tree is represented by a 16 bit vector, and bit encodings of parse trees are built up out of these vectors, Pollack trained the net using backpropagation to learn the underlying structural components associated with the input sentence. After training, the net was tested on novel trees corresponding to novel sentences containing the same constituents. The model succeeded in abstracting the relevant trees although mal-formed trees were occasionally generated.

Certain of the successful trials represented noun-verb combinations which were novel in the sense that the particular noun did not occur as an argument of the verb in the training corpus. This clearly suggests that the net succeeded in inducing some aspects of the compositional structure implicit in the training
set. However, Pollack's sole argument for systematicity rests on the fact that all 16 cases of (Loved X Y), with X and Y chosen from the set {John, Mary, Pat, Man} were reliably represented by the net even though only four of them were included in the training set. Although one of the words, man, never appeared as an argument of this particular verb in the training, all the words appeared as both subject and direct object of a verb somewhere in the training set. Thus Pollack can at best claim to have established a form of 'weak' systematicity.

Another problem is that the RAAM net was only trained on pre-parsed explicitly parenthesised structures whose syntactic structure had already been analysed by an external agent. The training regime is thus parasitic on the mediation of an intelligent agent acting as an arbiter. Simplification is acceptable given Pollack's goal of demonstrating RAAM's capacity for structure sensitive processing, but does not fit well with the long-term goals of connectionist theorising in cognitive science, namely to supplant the classical paradigm.

5.3.4 Smolensky's approach: Using tensor product representations

Smolensky approached the issue of whether highly structured, compositional representations can be realised in connectionist nets from a more mathematical perspective. Using tensor calculus as a tool, concatenative symbolic representations (including nested structures) can be translated into tensor product representations. A tensor product of two vectors \( \mathbf{v} \) and \( \mathbf{w} \) is the vector that results from the pairwise multiplication of each element of \( \mathbf{v} \) with each element of \( \mathbf{w} \).

Technically tensor-product representations are characterised as follows (see Smolensky, 1990 for details):

Let \( \zeta \) be a set of \( n \)-element vectors, each assigned to represent a 'filler' that can occupy various roles. Suppose \( R \) is a set of \( m \)-element vectors, each assigned to represent a 'role' that can be occupied by members of the filler set. Then for any two vectors \( \mathbf{r} \) and \( \mathbf{f} \) from \( \zeta \) and \( R \) respectively, the tensor product of \( \mathbf{f} \) and \( \mathbf{r} \) (i.e. the \((n \times m)\) element vector \( \mathbf{f} \times \mathbf{r} \)) represents a 'filler/role binding'. It is a representation of a particular filler occupying a particular role.

Given a set of vector representations of words as feature lists and a set of vector representations of syntactic roles such as subject, verb phrase, object, and so forth, the representations of sentences can be constructed as follows. The representation of a word, say John, in the role of subject is the tensor product of the vector representing 'John' and the vector representing the role of subject. The tensor product representation of the sentence is formed by vector addition of the tensor products representing each word in the sentence in its role in that sentence. If the vectors representing words and roles meet certain
mathematical conditions we can ‘ask’ a network what the subject of an actively represented sentence is by activating the subject-role vector. It will respond by activating the vector representing John. Likewise we can ask it what ‘John’ is in the sentence by activating the ‘John’ vector, and the subject-role vector will be activated. The same process can be followed for the other words in the sentence.

Tensor products provide a very clear illustration of the encoding of constituent structure by relative position on an activation landscape. Unfortunately, tensor product representations of complex structures necessarily involve many more units than the vector representations of the constituents of those sentences, making recursion difficult.

5.4 TRACKING COMMONSENSE REASONING

The examples presented in the previous section show that at the current level of ANN technology, these models have some difficulty in dealing with compositional, recursive aspects, and variable bindings associated with the syntactic aspect of language. ANNs can be developed to model structural properties, but the work is still largely programmatic. For this reason, many researchers have been somewhat pessimistic in their assessment of the potential of connectionist approaches to language and reasoning. As Norman (1986) points out ANN models are decidedly "low level" entities. The subsymbolic or microstructural characteristics of the processes and structures associated with these networks may be appropriate for theorising about aspects such as perception and conceptual structure, but seem much less applicable to typically symbolic activities such as language understanding and problem solving. Still, while there are undeniably some limitations in the use of ANNs as general tools for language processing at this stage, the assessment of their relevance presented by some classical researchers (e.g. Fodor and Pylyshyn, 1988) is unduly negative.

First of all, they ignore the fact that some aspects of ANN modelling have been innovative, presenting not only a new formulation of computational and cognitive processing, but an empirical methodology for exploring issues in language. For example, connectionist approaches make it feasible to examine some of the assumptions underlying the nativist position that are often just taken for granted. In a simulation of language learning, Elman (1993) discovered that learning is more tractable for a network if it starts with limited exposure to the pertinent data and is only gradually exposed to data of increasing complexity. This finding gives some support to an hypothesis about the role that ‘motherese’ may play in language learning. Motherese is assumed to be a restricted code which parents of young children spontaneously use in their interaction with their children when these are in the process of acquiring language (Taylor & Taylor, 1990, pp. 236-240).

Secondly, computational linguists usually adopt some or other formalism when they develop
computational theories of language. Logic is a favourite choice because it is a very expressive system which can be used to describe the semantic and syntactic properties of language often using proof-theoretic approaches. For instance, researchers in AI and natural language processing specialists working on dialogue systems concentrate on proof theoretic approaches. In dialogue systems used for communicating information about train destinations, researchers are interested in the form of a constructive proof that a person can get to a particular station, rather than in the mere truth of the proposition. The proof informs us how to actually get there, and a dialogue system needs this knowledge in its interaction with users (see Allen, 1995, pp. 485-489). One of the main problems inherent in these logic-based approaches is that they are difficult to apply to some everyday reasoning problems as Chater and Oaksford (1991) point out. Connectionist approaches, while not solving the problem of developing computational theories of everyday reasoning, do yield a new way of attacking the problem. In addition, connectionist approaches are at least sufficiently expressive to deal with some aspects of the semantic representation of information.

Two arguments emphasising some shortcomings inherent in the classical perspective about semantics are surveyed below. The first is an argument by Chater and Oaksford (1991) who claim that the classical 'logicist' tradition has difficulty in dealing with the patterns of defeasible reasoning that occur in human commonsense reasoning. The second considers role of efficiency in cognitive processing.

5.4.1 The problem of defeasible reasoning

The main problem that Chater and Oaksford (1991) address is that everyday reasoning is often defeasible, because humans regularly reason by default, or make default assumptions about their environment. The notion of a 'default' stems from Marvin Minsky's (1975) programmatic work on frame theory. Essentially a default can be thought of as the assumption that typical properties of a class of objects or events are inherited by a member of a class unless specific evidence to the contrary exists.

For instance, the statement Birds fly expresses a proposition that is true in general because flying is a property that we typically associate with members of the bird category. Unless we are explicitly told that a bird cannot fly, we assume it can. Our assumption that birds fly is a default assumption, and the inference that we draw with regard to a particular member of the class is a default inference because it is subject to error: it is typically, but not necessarily true. The inference is characterised as operating by default, because there are notable exceptions (e.g. penguins and ostriches) of birds which do not fly.

Such a default inference has the form “unless any specific evidence to the contrary is known, assume that X holds”, where X denotes the applicable default. Any conclusion sanctioned by a default can, however, be defeated as new information comes to light which conflicts with the initial inference drawn.

For this reason, default inferences are said to be defeasible or non-monotonic because they can be cancelled by subsequent information.
It is now widely recognised that such default assumptions play an important role in language understanding and cognition (Collins & Michalski, 1989). However, an important aspect of defaults, namely that they are defeasible, has proven difficult to account for in formal theories of language understanding and cognitive processing developed within the classical tradition. As mentioned above, defaults exemplify a pattern of reasoning based on typical properties of objects and events. It may not be immediately obvious why defeasible reasoning is a problem for classical, logic-based approaches, because this form of reasoning is so natural to us. It is actually a problem that only emerges clearly in formal approaches to language and reasoning, and it has been investigated mainly in symbolic, AI approaches. However, it also applies more generally to classical cognitive science as Chater and Oaksford (1991) argue, because classical cognitive scientists tend to adopt logic as a formalism (e.g. Cann, 1993; Barwise & Perry, 1983). The problem itself, and the solutions that have been proposed for it, are rather technical. However, it is relevant to present a brief introduction (without going into all the details) of some of these issues, because understanding why they crop up serves at least the purpose of showing that classical cognitive science is not a completely “healthy” science. Hence, in what follows a very brief sketch of the problem is provided. In first-order logic, a statement such as birds fly can be rendered as:

(1)  \( \forall x. Bird(x) \Rightarrow Flies(x) \)
(where as customary “\( \forall \)” denotes “for all” and “\( \Rightarrow \)” is to be read as “implies”)

Given the information that Peggy is a bird, we now infer that Peggy can fly. Next add the further statements:

(2)  \( \forall x. Penguin(x) \Rightarrow Bird(x) \)
(3)  \( \forall x. Penguin(x) \Rightarrow \neg Flies(x) \)
(In the above statement “\( \neg \)” stands for the negative operator “not”)

If we later learn that Peggy is a penguin, the above statements lead to a contradiction, because by (3) we infer that Peggy does not fly, whereas the conjunction of (1) and (2) entails that it does fly. What we need is the conclusion that (3) is somehow more specific than (1), and should prevail if we know for a fact that Peggy is a penguin. An obvious solution to the problem is to state explicitly that penguins constitute a subclass of birds that cannot fly. Hence, if it is known that Peggy is a penguin, the inference that it flies should be blocked. This idea is implemented below:

(4)  \( \forall x. Bird(x) \land \neg Penguin(x) \land \neg Ostrich(x) \land \ldots \Rightarrow Fly(x) \)
(in the above expression the “\( \land \)” stands for “and” so that the statement says that unless a bird is a penguin, or an ostrich, or .., it can fly)
The representation does indeed succeed in blocking the inference that Peggy flies if we are informed that it is a penguin, but additional problems now emerge. Note that we are required to list all the non-flying birds explicitly. The list is tedious to compile because there is a whole array of peculiar creatures in the set of non-flying birds, such as dead birds, newly hatched birds, birds with broken wings, birds in a cage, birds suffering from a fear of height, birds with their feet set in concrete, birds whose feathers have been plucked out and so forth. Furthermore, recall that if we only know that Peggy is a bird, we would like to infer (by default) that it can fly. Yet (4) requires us to first prove that Peggy is not a non-flying bird before concluding that it can fly, which is impossible if we only know that Peggy is a bird! In contrast, we would like to derive that Peggy can fly without first showing that it is not abnormal in any way. This task seems to lie beyond the scope of standard logic because the pattern of inferences involved is nonmonotonic.

The general problem reduces to the fact that the theorems of first-order logic are monotonic. If A and B are two first-order formulae, it follows that if a statement is inferrable from B, it is also inferrable from A if A is a superset of B. First-order logic guarantees monotonicity, because the theorems of logic are additive, they persist as new, additional information comes to light. Conversely, as we have seen, commonsense reasoning is often defeasible. We sometimes retract our inferences as new facts become available. Because what appears to be a plausible inference can be defeated by the addition of further information, the resulting pattern of reasoning is said to be nonmonotonic. Defeasible inferences are nonmonotonic in character, and thus fall outside the scope of first-order logic.

The problem is of course not insoluble because several non-monotonic logics have been developed to address the shortcomings of first-order logic applied to everyday reasoning (e.g. Reiter, 1980; McCarthy, 1980). However, so far most of these approaches have proven theoretically adequate, but not feasible in practice because they are subject to computational complexity limitations (Chater & Oaksford, 1991, p.104) - the concept of ‘computational complexity’, and its relevance to cognitive science, is discussed in the next section. Connectionist systems provide an alternative mechanism for developing approaches to everyday reasoning. As Chater and Oaksford (1991, p. 34) are quick to point out, ANN models are also subject to computational complexity issues, just like logic-based systems. However, ANN models have learning capabilities. Once a network has learned, it draws inferences as rapidly as it takes for the activation to spread from input to output units. Chater and Oaksford (1991, p. 33) suggest that in this sense the behaviour of ANNs mirrors the human case. Human learning may be a slow process, but inference becomes effortless once learning is in place. However, this suggestion fails to address the main dilemma that ANN modellers face. Although connectionist systems can simulate a form of default reasoning, this is restricted to specific instances. These systems do not ‘naturally’ incorporate a mechanism for variables binding, which is essential if ANNs are to be used for complex reasoning tasks. For example, without a mechanism for binding variables, ANNs are limited to inferences ranging over the instances that they have explicitly encountered. They cannot reason outside their training space.
(Marcus, 1999).

There have been attempts to endow connectionist models with such common sense inferencing capabilities. One way to deal with variable binding is to use temporal aspects of the units' activation, in addition to substituting the use of instantaneous activation values. Shastri and Ajjanagadde (1993) exploited phase synchronisation by allowing different phases in an activation cycle to represent different objects involved in reasoning, and representing variable binding by the in-phase firing of units. In the network the activity of the units oscillates in time, and reasoning is identified with the propagation of a pattern of rhythmic activity. An item in working memory is constituted by a phase within the pattern. The variable bindings needed for representation of propositions are marked by synchronous firing of a particular set of units (called "bound" units). Items of knowledge are represented by sub-networks, each representing a certain state or action (such as "walk: or "give") and units within the subnets represent objects used by the actions. Rules are represented by interconnections which determine the propagation of rhythmic patterns of activity, and particular units by filler nodes. The network yields an interesting solution to the variable-binding issue, but makes use of localist representation, and Shastri and Ajjanagadde (1993, p. 485) are pessimistic about using a purely distributed model for commonsense reasoning remarking that such a system "cannot have the necessary combination of expressiveness, inferential adequacy, and scalability".

A rather complex ANN system for commonsense reasoning was developed by Sun (1994; see also Sun, Merrill & Peterson, 2001), called CONSYDERR. It is put forward as an integrated model to deal with a set of problems of commonsense reasoning such as evidential combination, similarity matching, and inheritance (i.e. making inferences based on hierarchical conceptual relationships). In the model these problems are treated in a single, unified framework involving rule application plus similarity matching. The model consists of two networks grafted on one another. The one network (CL:CONSYDERR) contains domain specific information and performs similarity processing based on a feature analysis of the input it receives. The second network (CD-CONSYDERR) is trained to make educated rule-based guesses (i.e. these are soft, connectionist rules) based on the information it receives from CL-CONSYDERR. For example, if told that Peggy is a penguin, CL-CONSYDERR will apply its rule-based scheme to determine whether Peggy can fly based on Peggy's features. Sun (1994, p. 241) goes so far as to claim that CD and CL: "are not just implementations of their symbolic counterparts, but better computational models of human reasoning".

There are still many problems associated with ANN approaches to commonsense reasoning relating mainly to the limited expressiveness of the current generation of ANN models. In order to adequately model language, these systems need to be able to cope with the representation of the meaning of complex sentences, and as Section 5.3 showed, ANN researchers are now only beginning to construct systems capable of describing some aspects of sentence structure. In contrast, classical approaches can deal with
sophisticated aspects of sentence structure.

5.5 **EFFECTIVE VERSUS EFFICIENT COMPUTATION**

Much of the argumentation presented in favour of classical systems rests on the premise that they are computationally as powerful as can be, and consequently can cope with any aspect of human cognition which can be described in information processing terms. Recall from the discussion in Chapter 3, that any input-output mapping can be carried out by a Turing Machine, and by extension with a symbol system architecture of the kind analysed by Newell (1980). Moreover, the Church-Turing thesis suggests that this classical, symbolic level defines what computation is all about, so that it may be irrelevant to look further to alternative formulations of computation. Hence, according to this view, if ANN models are presented as new formulations of computation they may not be of much interest to cognitive science.

One problem with this conception of computation is that it presents what one may call a mathematical as opposed to an engineering conception of information processing. The focus is almost exclusively on effective computation, on what set of operations can effectively be computed by a system, and ignores the efficiency with which these computations are carried out. It therefore considers computation as any task or algorithm that can be executed by a given system in theory, leaving out the issue of whether the computations can be carried out in a practically feasible way. It is in this theoretical sense that a TM is an extremely powerful device, because it is a rather inefficient processor when viewed from a practical perspective. In order to show that the distinction between effective and efficient may be relevant after all, a little digression is necessary to give a brief introduction to the theory of computational complexity. To set the stage for the discussion, it is pertinent to consider an issue in cognitive science which suggests that efficiency plays a role in cognition. The issue is a problem called *The frame problem* which initially arose in the context for dealing with temporally related situations in AI.

McCarthy and Hays (1969) developed a simulation of a simple blocks world, with the aim of formulating a logical description of basic facts such as “The block is on the table”, and capturing inferences associated with changes to the world (e.g. if some blocks are moved around). In their approach, events in the world are stated as temporal changes taking place from one situation (frame) to the next. For each change in the situation, it is necessary to specify which facts change as a result of the particular action, and which do not change. These changes were formulated as ‘frame axioms’ which are basically first order logical formulae. Rather than presenting McCarthy and Hays (1969) discussion of the issues that crop up in the computational description of such a changing world, a rather more amusing depiction of the general implication stemming from this research is presented in a story borrowed from Dennet (1987). In the presentation I have changed the text, but the story and the moral contained in it belongs to Dennet.
Envisage a world, way in the future. In the world lives a robot called HAL100 by its creators. HAL100’s primary goal in life is to safeguard its own life, to survive in the world. One day it learns from its creators that its spare battery is locked in a room with a time bomb programmed to go off shortly. By applying its powers of deductive reasoning HAL100 comes to the conclusion that it is clearly to its own benefit to rescue the battery. After some reflection, it decides on a course of action, which entails first locating the spare battery, and then, as a second stage, rescuing it from the imminent threat. HAL100 quickly manages to find the room, unlocks and enters. It surveys the situation and formulates a plan. It notices a trolley in the room and reasons that by placing the battery on the trolley and executing an action called PULL-OUT(WAGON, BATTERY) the precious battery will be saved. HAL100 immediately acts and succeeds in pulling the trolley out of the room just before the bomb explodes. Alas the bomb was also on the trolley! The misfortunate HAL100 knew the bomb was on the trolley, but did not realise that by pulling the trolley it would transport the bomb along with the battery. It missed an obvious implication of its clever plan.

The designers are undeterred and immediately go to work on an improved robot. “The solution is obvious” they proclaim. Our next robot must be able to contemplate the intended consequences of its actions as well as their side effects. It must be able to derive the implications flowing from the descriptions it uses in formulating its plans. They design a more advanced prototype, ten times smarter than the now defunct HAL100 and call it the HAL1000 robot deducer. HAL1000 is confronted with the same predicament that led to the abrupt demise of its predecessor. Like the former HAL100 (but slightly quicker) HAL1000 sizes up the situation and concludes that the action PUT(WAGON, BATTERY) would be the initial action of its rescue plan. It then started to ponder the possible side effects of its actions. It had already deduced that the size, colour and shape of the room will remain unaltered, and had just succeeded in proving that the distance the trolley has to traverse is shorter than the cube of the sum of the distances between its four wheels, when the bomb explodes.

Slightly less enthusiastic this time, but not yet completely despondent the designers go back to the drawing board. After some brainstorming, they conclude that the crux of the problem was that the robot could not distinguish between relevant and irrelevant inferences. “We must design a new robot and teach it to ignore irrelevant inferences”, they decide. After some time they successfully develop a method for tagging inferences as either relevant or irrelevant to the task at hand, and program this method into their new prototype, the HAL2000, superior reasoner. HAL2000 is presented with the dilemma that resulted in the undoing of its two lesser predecessors. After a while the designers are surprised to find it sitting, hunched over, deep in thought, the “native hue of its resolution sickled o’er with the pale cast of thought” as Dennet (with Shakespearean verve) so eloquently puts it. “Do something you idiot the bomb is going to explode” they cry out in desperation. “I am” it responds. “I am ignoring thousands of implications that I have already deduced to be irrelevant. As soon as I find a new irrelevant implication, I add it to my list, and…” the bomb explodes.

The robot example was used by Dennet (1987) to illustrate the frame problem. However it also exemplifies the effect of processing time on computation. HAL1000 and HAL2000 both went astray because they started pondering the almost infinite set of consequences associated with even simple actions. In both cases the robots were possibly competent, but very inefficient reasoners. They could
not deal with a combinatorial explosion of possibilities when they started pondering the results of their actions. Thus the robot scenario portrays inefficient reasoners, but is also illustrates the role that problem complexity plays in reasoning. For example, drawing out all the possible consequences of an action and evaluating these proves to be ‘intractable’, even in Hays and McCarthy block world.

Intuitively dealing with a changing world may not seem such a difficult task to us humans, because we cope with a changing environment on a constant basis. If the postal delivery boy throws John’s newspaper into the garden every day, we know that when it rains he will wrap it up in plastic to protect it from the rain because the newspaper will be damaged when it becomes soaking wet. We also know that other items such as a hosepipe or a car need not be protected from the rain. The ‘knowledge updating’ task does, however, pose a problem for any computational account of the actual cognitive processes involved in coping with the world. The frame problem illustrates that in dealing with the requirements of a changing world, certain types of reasoning processes (e.g. those involving a description couched in standard first-order logic) may just turn out to be inefficient. These descriptions, and theories based on them, are computationally inefficient because they give rise to a combinatorial explosion of computations which cannot be satisfactorily carried out in practical situations. The descriptions may be quite effective because given enough time (and sometimes this may make enormous demand on resources as I will show further down) the cognitive task can be carried out effectively. The problem is therefore not that there is not a computational solution for such tasks, but rather that the proposed solution may not be practically feasible. Recall that in the previous section it was mentioned that the main problem with some of the default reasoning formalisms presented by AI researchers is that these formalisms cannot deal efficiently with common sense reasoning. For basically the same reason, Fodor (1980) holds that there is no adequate theory of ‘central systems’, because the process of general reasoning (see Chapter 3, Section 3.3.5) is “Quienian” in that it implicates the entire belief system, and that it therefore cannot be adequately captured by tractable, classical-style (i.e. algorithmic) explanations.

Oddly, even though the issue of computational efficiency is obviously of considerable relevance to information processing theories of cognition of the type elaborated in cognitive science, there is almost no discussion of this aspect in current cognitive science. There are fortunately two exceptions to this general omissions. Both Steedman (1998) and Frixione (2001) give good introductions to some underlying theoretical issues. The presentation given below gives a brief outline of the theory of computational complexity.

5.5.1 A short primer of computational complexity theory

Computational complexity theory (CCT) analyses the computational structure of problems (i.e. it views problems as information processing tasks). The assumption underlying CCT is that difficult problems
make heavier demands on processing time and memory resources than easy problems, and consequently that processing time is an indication of problem complexity. The type of analysis pursued is rather abstract. The goal of the theory is to map out the solution space of a problem (i.e. an information processing task) in terms of the demands it places on time and resources. Hence, according to this view, problem complexity is given by the amount of computational resources (mainly processing time) that will be expended in an attempt to solve the problem, and this time is expressed as a function of the input size. Considering the slope of this function gives an indication of how task difficulty increases with input size. The growth of this function is taken to be symptomatic of problem/task difficulty.

5.5.1.1 Some hard problems

One of the consequences emerging from the analysis of the growth of functions is that there are some problems which can be solved in theory, but not in practice, because they place excessive demands on computational time and resources. An example should help to make this clear.

Imagine that you have to plan an itinerary for a travelling salesman who is required to visit a number of cities. You are given a map on which the cities and the distances between them are indicated, and you are asked to find the shortest route that passes through all the cities and will take our salesman back to the starting point. One approach to this problem is simply to trace all possible routes, measure their distances and select the shortest one. Such an approach will yield a correct answer, but if the number of cities were to increase to 100, the problem will become completely intractable even on the most sophisticated computers. Simple mathematical analysis shows that the search for the shortest route is the strategy outlined above is equivalent to the computation of n! (n factorial), for different values of n depending on the number of cities to be visited. This is because the first city can be selected in n ways, the second in (n-1) ways, the third in (n-2) ways, and so on. The result is a combinatorial explosion n * (n-1) * (n-2) ... 1. For n = 10, there are 3 628 800 different routes to consider. For n = 40 this number grows to 8 159 152 832 247 897 734 345 611 269 596 115 894 272 000 000 000 000 (as computed using the variable precision arithmetic function in Matlab 5.0 - Math Works, 1997). For n = 100, the number of alternatives is unimaginably large, and for all practical purposes a determination of the shortest route for such a number of cities is computable.

Another example where no efficient solution can be obtained by exhaustive computation is the boolean formula known as Satisfiability or SAT. The problem is illustrated by considering an arbitrary boolean formula such as:

\[(x \lor y' \lor z') \land (y \lor z \lor u) \land (x \lor z \lor u') \land (x' \lor y \lor u)\]
and then to find an assignment of true and false for the variables which would make the expression true. SAT is considered to be a prototypically intractable problem. The only known solution to it is to progressively attempt every possible combination of assignments to see whether they yield the required result. On large formulas this approach becomes impractical because the number of possibilities increases exponentially. Two assignments (true/false) are possible for each variable so that the number of assignments increases with $2^n$ (an exponential function shown in Table 5.4 further down).

While intractable by deterministic means, SAT can be solved with the aid of an oracle performing a series of lucky guesses, thus recording the correct assignment for each variable. It may seem somewhat strange to define problem solving in terms of oracles, but this is quite a legitimate procedure applied to the class of very hard problems, for which systematic calculation of all the possible solutions is clearly unfeasible. The guessing approach can even be mechanised by developing a guessing computer (technically, a non-deterministic Turing machine), as shown in Cook (1971). For many hard problems, it is easy to verify whether such a specific choice of assignments for the variables in SAT yields a solution to the problem.

### 5.5.1.2 The growth rate of functions

A measure of computational complexity can be obtained by investigating the time and resources consumed by different information processing tasks. The table below (taken from Harel, 1987, p. 158) analyses the time taken by a computer running at a million instructions per second for varying input sizes (n) on tasks bounded by different functions. To interpret the table, note that each function describes the number of logical operations required to solve a given task. In the case of the function $n^2$, for $n=10$, this amounts to 100 operations. Dividing by 1 000 000 steps per second (the processing speed of the computer), yields 0.0001 seconds, which is the time that the hypothetical computer will take to perform the task.

<table>
<thead>
<tr>
<th>n/function</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>300</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n^2$</td>
<td>0.004 sec</td>
<td>0.025 sec</td>
<td>0.01 sec</td>
<td>0.09 sec</td>
<td></td>
</tr>
<tr>
<td>$n^3$</td>
<td>0.1 sec</td>
<td>3.2 sec</td>
<td>5.2 min</td>
<td>2.8 hours</td>
<td>28.1 days</td>
</tr>
<tr>
<td>$2^n$</td>
<td>0.001 sec</td>
<td>1 sec</td>
<td>35.7 years</td>
<td>400 trillion years</td>
<td>a 75-digit number of centuries</td>
</tr>
<tr>
<td>$n^4$</td>
<td>2.8 hours</td>
<td>3.3 trillion years</td>
<td>a 70-digit number of centuries</td>
<td>a 185-digit number of centuries</td>
<td>a 728-digit number of centuries</td>
</tr>
</tbody>
</table>

*Table 5.4 The growth rate of some functions*
It is instructive to bear in mind that the big bang is estimated to have occurred about 15 billion years ago. Thus if our hypothetical computer had been busy working on a task subsumed by the function \( n^k \) for \( n \geq 20 \) from the inception of time, it would still not have finished its computations in 2001! Using a computer as much as a billion times faster will not affect the processing time significantly for large values of \( n \), nor will parallel computation make a difference. Rather it appears that some problems are intrinsically hard, and no amount tinkering or speedup is likely to make them more tractable.

### 5.5.1.3 Elements of complexity theory

One of the objectives of CCT is to classify problems in terms of their inherent difficulty. To understand how this classification scheme works, consider again the growth rate of the functions shown in Table 5.4 for increasing values of \( n \). Notice that some functions (e.g. \( n^2 \)) are fairly well behaved in that the time taken by the computer to execute such problems grows linearly with increasing problem sizes. However, other functions (e.g. \( 2^n \)) increase exponentially. CCT abstracts from results such as these to divide problems into two main categories:

- **P**: Problems solvable by a deterministic Turing Machine in Polynomial time (time \( n^k \) for any integer \( k \), where \( n \) denotes the problem size).
- **NP**: Problem solvable by a Non-deterministic Turing Machine in Polynomial time. Informally a problem is in NP if its answer can be guessed, and the correctness of the guess verified, in polynomial time.

The NP problems are prototypically difficult problems because their complexity (i.e. solution time) grows exponentially. They can be solved effectively but the number of computations involved would be exorbitant for large values of \( n \), so that they cannot be solved efficiently. Some problems are even harder than the class of NP problems. These problems fall into classes described in terms of polynomial and exponential space (unlike time, space can be re-used during computation). These very hard problems fall outside the scope of this short exposition of CCT, but it is interesting to note that certain generative linguistic problems fall into classes beyond NP (see Barton, Berwik, & Ristad, 1987). Informally, these problems can be characterised in terms of double or triple exponential functions so that they grow much more rapidly with increasing input sizes than even the exponential functions shown in Table 5.4. The very hard category of problems are of somewhat academic interest, most practical problems fall in the categories of P and NP.

Complexity classifications are established using a proof technique known as reduction. A reduction typically proceeds by converting a problem \( P_1 \) of unknown complexity into a problem of known complexity \( P_2 \). The reduction itself is effected in polynomial time. The interesting consequence is that
if an efficient algorithm is found to solve $P_2$, the same algorithm can immediately be extended to $P_1$, simply by converting instances of $P_1$ into $P_2$. Problems in $NP$ which represent the class in this way are known as $NP$-complete problems. The set of $NP$-complete problems are, in a sense, the hardest problems in $NP$. An example of an $NP$-complete problem is SAT, which was described above. There is some uncertainty as to whether an as yet undiscovered polynomial algorithm exists for the $NP$ class of problems. The quest for such an algorithm has occupied mathematicians since 1971, and is one of the most important unresolved issues in theoretical computer science. Note that if such an algorithm were to be discovered then $P=NP$. However, most mathematicians doubt that such an algorithm exists, and strongly suspect that $P \neq NP$. Thus, for all practical purposes the $NP$ class of problems can be regarded as intractable by deterministic means.

5.5.1.4 Complexity issues in cognitive science

The main point for the digression into CCT is to show the role that computational complexity can play in practical applications of information processing approaches. In the classical paradigm, emphasis is placed on the power of a computational system, but the efficiency with which the system carries out the task is usually ignored. The explanations associated with some classical cognitive theories may therefore pose effective solutions to cognitive processes but their solutions are not necessarily efficient. For example in Janeke (1997) it is shown that an explicit theory of counterfactual reasoning proposed in the language of first-order predicate logic is computationally intractable. Likewise, Reiter (1980) presents a very elegant solution to the problem of default reasoning, but his logic is undecidable in principle and intractable in practice, because deciding whether a default rule applies involves consistency checking, which is an NP-hard problem (see Garey and Johnson, 1979).

Classical approaches often rely on rule-based formulations of cognitive processes (e.g. Chomsky, 1980) which can give rise to excessive computations because rules may interact with one another leading to combinatorial increases (see Barton, Berwik & Ristad, 1987). In some of the more recent approaches in generative grammar, the role of rules in cognition is downplayed. Thus, Atkinson (1992) shows that in ‘principles and parameters’ theory, a child’s task in learning language is construed as that of fixing parameters that constrain the learning task. During learning the child discovers, for example, that the particular language is left-branching and this greatly facilitates the learning task (see Atkinson, 1992).

In the classical, symbolic approach, language processes are typically conceptualised in terms of rule-based procedures (Bechtel & Abrahamsen, 1991) and aspects relating to efficiency and computational resources are eschewed. It is generally assumed that these aspects pertain to implementations of classically developed theories, as suggested by Broadbent (1985) in his critique of McClelland and Rumelhart’s (1985) memory model discussed in the previous chapter. The main point underlying my
presentation of CCT is to show that such a rigid distinction between theory and implementation may not be feasible in the cognitive sciences. If one adopts an information processing perspective on cognition then the practical feasibility with which cognitive algorithms are to be carried out, must be taken into consideration.

In contrast to rule-based accounts of cognition, many ANN models are formulated as constraint-satisfaction systems and are therefore tailored to deal with aspects of computational complexity, as exemplified by Hopfield and Tank's (1985) approach to the travelling salesman problem. Likewise some applications of ANN ideas to language processing reflect a progression to a constraint-based formulation of linguistic processes. For example, Prince and Smolensky (1997) show that a version of harmony theory (see Smolensky, 1988) containing non-numerical constraints makes it possible to order morphological and phonological properties of different languages in a dominance hierarchy. In terms of this position it becomes possible to consider the way in which particular languages evolved, and the learning tasks associated with them can be viewed as essentially a resolution of a set of parallel constraints. The net effect is a conception that has progressed from viewing language as a sequence of rewrite rules (implied by early generative linguistics) to a theory of constraints, intended to render language processing tasks computationally efficient.

5.6 CONCLUDING COMMENTS

As the two above arguments illustrate, the classical approach is not without some serious difficulties of its own. The critique levelled at ANN models by classical theorists should therefore be seen in the light of the controversies about the adequacy of classical models in providing a satisfactory way of coping with commonsense reasoning, and yielding an efficient, as opposed to just effective, computational theories. In the light of these shortcomings of classical theories, it would be premature to reject ANN approaches at this stage, even if they have some limitations currently in coping with aspects of cognition such as language and reasoning. For this reason Chater and Oaksford (1990) have suggested that in some respects Fodor and Pylyshyn’s (1988) critique of connectionism downplays the relevance of implementation in cognitive science. As already noted in Chapter 4, Fodor and Pylyshyn reject ANNs as implementational models, but implementational issues are only irrelevant to cognitive science if the algorithmic and implementation levels are completely autonomous. However, as the argumentation in the previous section shows theories presented at the algorithmic level need to take the efficiency with which the algorithm can be performed into account, and should pay heed to the actual constraints imposed by the physical system, such as memory limitations. Given that these levels of explanation interact in this way, implementation cannot be regarded as irrelevant to cognitive researchers. For similar reasons, Christiansen and Chater (1999) state that a form of ‘leaky recursion’ should be tolerated in theories of human language processing, because purely formal accounts of recursion make unrealistic
demands on memory processes.

Given the fact that ANNs might serve as implementational level descriptions, sensitive to the human cognitive architecture, they have some role to play in cognitive theorising. But what exactly is the status of connectionist theories of cognition, and how do they enhance our understanding of cognitive mechanisms? In the next chapter these issues are considered.
CONNECTIONIST MODELS
AND COGNITIVE THEORY: AN APPRAISAL

In the discussion so far, connectionist research has been placed relatively firmly within the information processing paradigm. In the context of this paradigm, it is typically assumed that connectionist approaches can be analysed as a species of cognitive architecture that is driven by a brain-like computational mechanism. In short, ANNs were viewed as neurocomputational systems, and it was argued that they differ from classical symbolic computational systems mainly in terms of the specific information processing approach they instantiate, one that is founded on distributed, subsymbolic, and parallel processing properties. However, the overarching assumption that connectionist nets are computational devices remained largely intact in the perspective on ANN modelling sketched so far. It is now appropriate to consider some of the issues surrounding the computational conception of ANNs in more detail, and to consider the theoretical relevance of these models in cognitive science.

Proponents of the classical view adopt a conception of cognition in which cognitive processes (and structures) resemble linguistic strings. On this view, the key property of conceptual structures is that they are sentential constructs which have both a syntactic function (determined by their placement in the string) and a semantics (i.e. a meaning). This type of approach is reflected in well-developed semantic theories such as model-theoretic semantics (see Cann, 1993), as well as in some AI approaches (e.g. Nilsson, 1998). As discussed in the previous chapter, an important assumption underlying most of these classical conceptions of high order cognition is that language and reasoning draw from sentential-type, compositional representations. In other words, it is assumed that the meaning of complex structures can be derived from the combination of their semantic units. In the classical conception of cognition these units are symbolic entities, they have semantic content and can therefore be used to compose more complex meanings. The classical theory is based on a well-developed semantic theory which emphasises the role of symbols and constituent structure. This classical computational paradigm has been relatively successful in AI tasks and in natural language processing applications (e.g. dialogue systems - see Allen, 1995).

The connectionist, ANN-based account has more difficulty to explain the semantic component of sentences. As the previous chapter showed, most of the interesting work done on connectionist approaches to language focused on word meaning and speech processing (e.g. grapheme-phoneme conversion), but applications of ANNs to traditional NLP topics such as sentence processing and
language generation have been much less successful. For this reason, some cognitive scientists such as Marcus (1998) and McCloskey (1991) have questioned the feasibility of the connectionist approaches to the syntactic aspects of language. Given the apparent limitations of ANN models in coping with some aspects of language, questions can be raised about their theoretical value in cognitive science - this is the main theme addressed in this chapter.

6.1 POsing THE PROBLEM

Keith Stanovich (1998, p. 25) jokingly put forward the following ‘theory’ of cognition.

> I have discovered the underlying brain mechanism that controls behavior. You will soon be reading about this discovery (in the National Enquirer, available at your local grocery). In the left hemisphere of the brain, near the language area, reside two tiny green men....And, well, to make a long story short, they basically control everything. There is one difficulty, however. The green men have the ability to detect any intrusion into the brain (surgery, X rays, etc.), and when they do sense such an intrusion, they tend to disappear. (I forgot to mention that they have the power to become invisible).

Stanovich’s ‘theory’ is intriguing, but does not quite meet the standards of explanation associated with current scientific practice (see Hawking, 1988, pp. 9-10; Green, 1999, pp. 364-370). Simply postulating that “tiny green men” manage everything, eschews the real hard task of spelling out the finer details associated with the operation of the mind. As an established research programme with at least a measure of scientific legitimacy, cognitive science aims to elucidate the mechanisms underlying cognitive functions, and this entails peering inside the proverbial black box to yield insight into the dynamics of human cognition (Pinker, 1997). Cognitive theories are required to clarify underlying processes, but they should also be formulated in such a way that they can be evaluated. Evidently the ‘tiny green men’ theory does not satisfy either of these conditions.

The questions I want to consider in this chapter is to what extent ANN theories of cognition are consistent with the mechanistic approach characteristic of cognitive research in the cognitive science framework, and whether they are not just sophisticated versions of the tiny-green-men style of explanation? As already suggested in the previous chapters, an extended polemic has arisen around ANN approaches to aspects of higher level cognition such as language, and some cognitive researchers have been very critical of ANN modeling. Thus, Oliphant (1997) bluntly remarks, paraphrasing a remark attributed to Goebels, that “Whenever I hear the word ‘connectionism:’ I reach for my critical faculties”. Further on in the same paper he observes that the link between psychology and connectionism may be disastrous, because:

158
The word "Psychology" does after all, begin with a "sigh" and end with "oh gee, why?". And "Connectionism" begins with a "con" (Oliphant, 1997).

In contrast, some authors have presented and evaluated ANNs in extremely positive terms. Allman (1989) even calls them "Apprentices of wonder". The moral of the story is that there are pros and cons to the use of ANNs in cognitive science, one should certainly not be blind to their faults, on the other hand it would be a mistake to ignore the potential of these systems in making a contribution to what Smolensky (1988, p.5) refers to as the "somewhat impoverished repertoire of cognitive science".

At the implementation level ANNs are automated curve-fitting procedures. For example, Sejnowski and Rosenberg's (1987) NETtalk simulation has a very large set of connection weight parameters (more than 18,000 weight coefficients) which were fit by an automatic learning rule. Few researchers in psychology or cognitive science would consider any analysis based on such a large set of free parameters to be a serious theory. Connectionist models are clearly proficient at capturing input-output mappings, but the value of the theory is not determined by the goodness of fit, but by the nature of the theoretical formulation. Once a successful quantitative model (connectionist, regression, or otherwise) has been formulated, this still does not does not mean one has explained the cognitive phenomenon concerned. A researcher is also required to provide a theoretical justification of the model and an explanation of the terms contained in it, of how these terms map onto the phenomenon that is being explained, and so forth.

In artificial intelligence, various different models have been put forward to explain default reasoning, but these relied on different logical formalisms, and can therefore be said to offer different explanations of the underlying processes (see e.g. McCarthy, 1980; Reiter, 1980). The problem is that researchers in this domain are concerned with formalising basic principles, and not enough empirical data are available to evaluate competing models (Ginsberg, 1993, pp. 212-228). The same problem seems to be occurring in the case of connectionism, where different models have been put forward to explain the same phenomenon.

A case in point is an influential model developed by McClelland (1995) which is intended to simulate the development in children's understanding of a puzzle called "the balance beam problem". The puzzle derives from cognitive developmental studies conducted by Inhelder and Piaget (1958) and Siegler (1981) in which they showed children a balance beam with different weights placed at varying positions from a fulcrum, and then asked the children to judge whether the beam will tilt when it is released, and if so in what direction. To solve various forms of the puzzle, children must attend to two dimensions at once (i.e. the particular weights placed on the beam, and their distances from the fulcrum). Developmental researchers have found that children appear to pass through a series of stages in their answers to the puzzle, and have postulated that these stages reflect a process in terms of which children acquire increasingly more complex rules. During these stages, children are thought to gradually integrate information about the two dimensions of distance and weight, in their judgments about the behaviour
of the beam.

In a first stage children appear to rely exclusively on weight and ignore information relating to distance. In this stage the heavier object is always assumed to tilt the beam. In a second stage children incorporate distance information, but only when the weights are equal. If the weights are equal, children take distance from the fulcrum into account, judging the side whose weight is the greatest distance from the fulcrum to go down. In the third stage children begin to incorporate information from both dimensions, but get confused as to whether distance or weight will determine outcome. Thus in situations when exactly the same configuration of weight and distance occur, they sometimes judge weight, and sometimes distance to have the decisive effect on the beam’s status. They only make consistently correct judgments in a final stage when they can evaluate the interaction between distance and weight. Siegler distinguished the use of four rules characterising children’s learning of the balance-scale problem. These are:

Rule I: only consider the dominant dimension.
Rule II: consider the subordinate dimension if and only if the values on the dominant dimension are equal
Rule III: consider both dimensions; in case of a conflict, muddle through
Rule IV: consider both dimensions in a proper way.

McClelland (1995) developed a connectionist model to simulate this process by which children develop insight into the operation of the balance beam. As the figure below shows, the network consisted of 20 input units, 4 hidden units and 2 output units to indicate the direction of tilt.

![Network Architecture](image)

**Figure 6.1** A sketch of the network architecture McClelland used to simulate the balance beam problem.

The model tries to simulate the way in which children acquire an understanding of the effect of using different weights and placement of weights on the balance beam. Input units were used to encode the
weight placed on the beam as well as the distance on which the weight is placed to the left and to the right of the fulcrum as the figure illustrates. In this way the connection structure of the network was constrained because input units that represent different dimensions of the problem (i.e. weight and distance) are connected to distinct hidden units. In the model, the activation patterns of the input units represent the perceptual stimuli about the configuration of the scale and the activation of the output units represents the judgment about its tilt. The network handles the balance scale problem by the following procedure:

☐ Represent the values of weight and distance of the blocks at both sides of the fulcrum on the input layer by activating units.

☐ Compute the activation of the hidden units and the output units.

☐ Translate the activation of the output units into a judgment (i.e. left down, right down, or balance).

The network was trained over many epochs (i.e. training cycles). During each learning phase it 'judged' 100 problems, and then received feedback so that the connection weights can be adjusted using the backpropagation algorithm. At both sides of the fulcrum, the maximum values of weight and distance are 5, which implies that there are $5^4 = 625$ different problems. The problems that were used during training were chosen randomly from this set of all possible problems, but in such a way that weight (the dominant dimension) is more frequently available for predicting outcome than distance (the subordinate dimension). McClelland (1995) called this the "environmental assumption". Each learning phase was followed by a test phase consisting of 24 problems, four of each of the six problem types.

McClelland reports that in 85% of the epochs the network responded consistently with Siegler's four rules. In comparison, Siegler (1981) showed that 90% of the children in the study fit the criteria. The network passed through the rules in the expected sequence from Rule I to Rule IV. However, unlike a group of 12-year-olds (15%) and a group of adults (30%) it never attained the final rule without regression to Rule III. During the development of the network, the classification remained consistent for some period, whereupon changes (e.g. from Rule I to Rule II) appeared suddenly. From these observations McClelland concludes that it shows a stagewise development. In addition, McClelland (1995, p. 176) suggests that the model's performance simulated the process by which children acquire an understanding of the weight*distance interaction, progressing from no Rule at all to a relatively stable performance on Stage 1 and Stage 2, and approximating Stages 3 and 4 after a period of vacillation (see McClelland, 1995, p. 176).

The model has been acclaimed widely as an innovative application of connectionist modelling (e.g. Schultz, Schmidt, Buckingham & Mareschal 1995). Subsequently, Schultz et al. (1995) developed a
model of the same task, but using a cascade correlation algorithm developed by Fahlman and Lebiere (1990). The cascade correlation learning procedure forms part of what Quartz (1993) calls constructive networks. Cascade correlation networks do not have a predetermined internal structure, but grow their hidden layers during training on the data domain, adding new nodes successively according to the residual approximation error. The algorithm succeeds in giving structure to a net, and often reduces the training time necessary for a given application. Such constructive approaches are now used in many ANN modelling tasks, because they enable a researcher to circumvent the difficult task of determining network structure, and also because they are judged to be more neurally plausible than the standard backpropagation procedure (Quartz & Sejnowski, 1997). The point at issue though is that the cascade correlation architecture is different from the backpropagation approach used in McClelland (1995), so that the question can be raised about the theoretical status of the two ANN simulations of the same cognitive task. How does one evaluate their respective contribution to cognitive theory? Are they best viewed as two different versions of the same theory, or should they be regarded as two different theoretical conceptions of the same cognitive task? If they are two different theories, how does one decide between them? If they are simply different versions of the same theory, how does one specify the underlying theory, and what then does a real connectionist theory look like? Unfortunately, questions such as this still lack satisfactory answers, so that at this stage one is left with the simple observation that the two simulations of the same phenomenon highlight the fact that connectionist architectures offer many degrees of freedom for the development of models. One learning rule can create many different networks - for instance, containing different numbers of hidden units - that each compute the same function. Each of these systems can therefore be described as a different algorithm for computing that function. Unfortunately there are as yet no a priori criteria available for deciding which of the possible ANN algorithms and architectures yield the most plausible cognitive theory of the phenomenon being studied.

Because very different ANNs can model the same phenomenon, Massaro (1988) maintains that they are too powerful to be scientifically meaningful. He argues that because their computational power appears to be more or less unconstrained, it undermines their explanatory value in cognitive science contexts. After all, if they can give rise to any behaviour (human and not human) they make poor models of human learning. For this reason some researchers advocate the use of highly structured models (Feldman, Fanty & Godard, 1988). If such a significantly constrained model were to match human data, it would be of more interest than the success of a general purpose mechanism because it would suggest that the human process under study is consistent with the constraints built into the model. The key to such a constrained design is probably to build in structures motivated by neurological, psychophysical and cognitive research. Several of the new models that have been proposed about human memory mechanisms (e.g. McClelland, McNaughton, & O'Reilly, 2000) reflect a constrained architecture, in which the design of network topologies are dictated by empirical findings about the neural mechanisms
underlying memory. Regier (1996) also made use of a constrained connectionist architecture to explore the way in which the spatial semantics of a set of open-class words (mainly function words such as prepositions) can be learned during early language acquisition.

6.2 CONNECTIONIST MODELS AS EMBODIMENTS OF COGNITIVE THEORIES

As indicated above, some researchers have taken a critical attitude in evaluating the theoretical relevance of ANN models. Oliphant's (1997) main point is that ANN models are associated with too many degrees of freedom, a point which has also been stressed by Massaro (1988). ANNs are powerful at learning and fitting input-output patterns, but the value of their contribution as a theoretical option in cognitive science should be found in their explanation of the phenomenon or task concerned; as noted above, having a quantitative fit per se does not guarantee that an ANN model makes a theoretical contribution.

McCloskey (1991) forces this point home using the example of Seidenberg and McClelland's (1989) model which was described in the previous chapter. Recall that the model is intended to simulate the cognitive processes implicated in converting orthographic representations into graphemic representations. The main goal was to show that a domain general learning architecture can account for the reading process, where traditional theories postulate a dual-route cognitive architecture. The authors argued that the network performs well at lexical decision and naming, and also exhibits several specific phenomena associated with studies of human subjects (e.g. an interaction of word frequency and regularity in the response latencies when subjects are required to name words). The issue to establish is in what way a network of this sort actually contributes to our understanding of human cognition?

Suppose that the Seidenberg and McClelland (1989) model is given the input I COULD DANCE, and upon processing it, the network generates the following output:

Word/Nonword Classification (116 msec)
phonological Representation /ay Kud daens/ (141 msec)

Suppose further that after the network is extensively tested on a broad variety of such inputs, containing both word and non-word stimuli, it is found that its performance matches the performance of human subjects reasonably well. Such a result would be impressive, but would it, as a black box, constitute a theory? Would it offer any explanation of how humans recognise and name words, or of specific phenomena such as frequency-regularity interaction? Before crediting the model with yielding an explanation of the psychological processes involved in word processing, one needs insight into the actual operation of the black box, and questions such as what aspects of its structure and functioning are relevant to the modelling of human performance need to be addressed. Moreover, this description would have to be in a form that made clear how the device was able to generate pronunciations, make word/non-
word decisions, and how its functioning gives rise to particular phenomena. One might then consider the description a theory of word recognition and naming, and the device a formulation of the theory.

The point is simple and obvious. Although the utility of connectionist networks (or other computational devices) to reproduce aspects of human cognition is intriguing, this ability does not qualify the network as a theory, and does not amount to an explanation of the relevant cognitive process. A particularly uncharitable perspective on this type of ANN modelling is mentioned in Regier (1996, p. 48), who jokingly puts forward the notion that it may be more enlightening to simply “beget a child” than to build an ANN model. After all:

"..your child will certainly exhibit the cognitive function you are interested in - more accurately than the model, for that matter - and then rather than empirically observe the model, you may observe the child."

The point is, of course, somewhat frivolous, because ANN research in the cognitive science context is not only aimed at exploring human cognition, but also at producing intelligent, human-like machines. There is considerable technological gain associated with the latter object, and ANNs play an important role in this artificial brain building endeavour (see e.g. Jani & Levine, 2000; Watanabe, Nakanishi, & Aihara, 2001). It is largely for this reason that AI researchers experiment with different network structures and learning rules.

Influenced by the issues mentioned above, some researchers have questioned the view that connectionist models should be viewed as theories on the grounds that they fail to satisfy the requirements of a good cognitive theory (McCloskey, 1991). In order to count as theory of a cognitive phenomenon an ANN must describe the phenomenon at a level more abstract than the particular network simulation, and answer questions bearing on issues such as how the cognitive function is executed and how specific phenomena associated with the cognitive domain arise. Such a formulation would include:

- organising and making sense of the available observations and allowing generalisations to be stated;
- providing clear-cut criteria for blame assignment in which both the success and failures of a prediction are taken into consideration;
- showing that the network models mechanisms are dictated by the general theory of the phenomenon in question.

In evaluating the theoretical contribution of a net one should therefore look beyond the fit and seek explicit information on the type of qualitative phenomena that the net predicts. It should be borne in mind, however, that theories are put forward by scientists and not by models. Simulation models are
powerful tools that help researchers develop, test, present and demonstrate the plausibility of their theoretical ideas. They do not, however, 'discover' the theory for the researcher, and only embody aspects of the general theory. ANN modelling is a complex task, and because the principles of computation in connectionist (parallel distributed processing) architectures are not yet well understood, a large part of the discovery process comes from working with the models themselves and experimenting with various architectures, input representations, learning rules and parameters, and so forth. The end product of the synergistic interaction between modeller and model is not just the model (or models) of a specific cognitive phenomenon, but a scientific article, in which the researcher's theoretical ideas, and their justification, are articulated.

Often in the presentation of such assessments of these models by the researcher, they are treated as experimental 'subjects', whose essential computational properties are inferred by the modeler. A good example is Plaut and Shallice (1993), who systematically investigated various assumptions underlying the simulation model of Hinton and Shallice (1991) - discussed in the previous chapter - which was developed to explain the co-occurrence of visual and semantic errors found in deep dyslexia. As Plaut (1995) explains:

The design issues included the definition of the task of reading via meaning, the network architecture (i.e., the numbers of units, their organization into layers, and how these groups are connected), the training procedure, used for adjusting connection strengths, and the procedure for evaluating the behavior of the trained network in its normal state and after damage. The major finding was that the occurrence of the qualitative error pattern was surprisingly insensitive to these detailed aspects of the simulation. Rather, what appeared critical was a more general property that all of the implementations shared: that units learned to interact in such a way that familiar patterns of activity over semantic features -- corresponding to word meanings -- formed STABLE ATTRACTORS in the space of all possible semantic representations (Plaut, 1995, pp. 299-300 emphasis in original).

It is interesting to note that Plaut and Shallice's (1993) case study occupied the complete 10th edition of the journal *Cognitive Neuroscience*. The model represents an elaborate exploration of aspects of deep dyslexia based on a complex attractor model. Moreover, the model makes some unique predictions. For example, it predicts a dissociation between cases which primarily involve word-concreteness effects accompanied by semantic errors and cases which are dominated by visual errors which are much less effected by word-concreteness. As Plaut and Shallice point out in their discussion, this is just the type of dissociation that has been observed in some deep-dyslexic subjects, and which has been difficult to explain using symbolic approaches. In contrast, such patterns of responses to damage occur naturally in the ANN simulation model, lending some credence to its viability as a basic model of reading and reading impairment. It is important to note that Plaut and Shallice's (1993) the model is not just data driven, but based on theoretical postulates such as that early visual processes in reading are input driven,
but that the production of semantic responses involve a dynamic attractor process. These and other theoretical postulates about cognitive aspects of the reading process are built into the architecture of the model itself. The ANN is not just a black box which has to learn everything from scratch. In fact, Plaut and Shallice’s (1993) simulation study shows that concepts such as stable attractors, basins of attraction, clean-up processes, superposition, and trajectories in weight space offer new and valuable ways of understanding how networks, and arguably how people, perform cognitive tasks. Whether or not they are right, it should be clear that the building blocks of connectionist explanations of cognitive performance are not units or weights, but theoretical, higher level descriptions of the functional properties and dynamics of networks during their learning and processing of information.

6.2.1 Using ANNs to test cognitive theories

Some researchers would argue ANNs may not be interpretable as theories in their own right, but that they nevertheless yield a methodology for testing and evaluating theories or theoretical assumptions. Kosslyn has been adamant that his main interest in ANN modelling is to use them to test the computational assumptions underlying theories of cognition, and that ANNs provide a sort of “poor man’s” way of evaluating the computational complexity of a given theory. He maintains that there are only two reasons for using such models, to determine how difficult a particular input-output mapping is, and to discover what information in the input is needed to achieve the mapping (Rueckl & Kosslyn, 1992, pp. 249-250).

This role of ANNs is exemplified by Kosslyn and his colleagues (Baker, Chabris & Kosslyn, 1999; Kosslyn, 1994; Kosslyn, Chabris & Baker, 1995) who assessed the efficiency of using a split network against a single network for processing spatial information. The general question they were interested in is whether the brain contains separate systems for processing the location and the shape of an object, based on an hypothesis that there are two distinct subsystems within the visual-spatial processing pathway (the dorsal system). One subsystem encodes categorical spatial relations, which describes the location, orientation, or other spatial characteristics of one object relative to another in terms of a broad equivalence class. The other subsystem encodes coordinate spatial relations, which specify an object’s spatial attributes with respect to another object in terms of precise metric coordinates (Kosslyn, 1994). A connectionist model was constructed to test the ease with which the two different types of mappings can be learned.

The use of an ANN to test the computational feasibility of a cognitive theory, is also illustrated by Rumelhart and McClelland’s (1986) model of the acquisition of the English past tense. The dual-route model of past tense learning and production may be viewed as a little theory. Now suppose someone suggests that a single route would be more parsimonious. The first problem such a counterproposal faces is how it is going to explain the U-shaped path of past tense learning. A supporter of the dual-route model
might argue that single route just does not have the resources to explain the data, and would ask how else (than the connectionist model) the U-shaped pattern of past tense learning should be explained. Such how-else arguments are prevalent and powerful in cognitive science, because if a given theory has no competitor then that theory will clearly prevail as the best explanation of the phenomenon. Rumelhart and McClelland's (1986) single route connectionist model of past tense learning demonstrates that single route models cannot be ruled out by the how-else argument. Rumelhart and McClelland's model was heavily criticised, but other ANN models have been proposed which addressed many of the objections (e.g. Plunkett & Marchman, 1991). Moreover, these models also inspired new empirical investigations, and generated predictions which could be tested (Marchman, 1993), leading in turn to further modelling.

In many respects running a good ANN simulation is like running any good experiment. The researcher starts with a problem or research goal, puts forward a hypothesis, a simulation as a means of testing the hypothesis, and a plan for evaluating the results of the simulation study. There does not seem to be any reason in principle why this approach exemplifying the typical connectionist research method of data → hypothesis → design of simulation → evaluating results of the simulation should be regarded as unscientific.

6.3 A DEFLATIONARY VIEW OF CONNECTIONISM

Even used in the restricted theory testing manner, ANNs give rise to some controversies. The main problem is that the mechanisms underlying the mappings discovered by ANNs are in many cases opaque, and difficult to analyse. For this reason some researchers hold that even as “theory testers” the contributions of connectionist algorithms to cognitive science is somewhat suspect, for three reasons (see e.g. Marcus, 1999; Green, 2001):

- The ability of such models to capture the right empirical generalisations can be challenged;
- The models are often exceedingly difficult to interpret, which mitigates their explanatory usefulness as computational descriptions of psychological processes;
- In many cases the functional architecture of these networks is not completely specified.

However, these allegations are somewhat overstated and it is possible to find examples that refute each point (see e.g. McClelland, 1988, p. 116). ANNs have already been of value to cognitive researchers because they have:

- led to new interpretations of basic phenomena in the literature as suggested by the ANN models of reading and reading impairment (e.g. Hinton & Shallice, 1991; Plaut & Shallice, 1993);
- provided unified accounts of what had previously been seen as highly disparate or even contradictory phenomena, as illustrated by Plaut and Shallice (1993) study discussed in the previous section;

167
clarified the relevance of computer simulation studies for adjudicating basic questions about the nature of the information-processing aspects underlying human cognition.

In addition, as argued during the previous chapters, connectionism offers a new way of thinking about cognitive phenomena, yielding an alternative set of conceptual and computational tools that should not be judged in terms of whether they are 'right' or 'wrong,' but rather, in terms of how useful they are for capturing those aspects of cognition that symbolic approaches experience difficulties with (e.g. content-based memory retrieval). Furthermore, some of the work done in the context of the connectionist paradigm can be seen as a challenge to the assumptions that some classical researchers make about the realistic status of their theories. Some connectionist researchers such as Churchland (1995) and Smolensky (1988) contend that classical accounts do not reflect actual 'theories in the head', but that they are merely instrumentalist descriptions. According to them, the proper account of cognition is given by explanations of the dynamic properties of connectionist models. Thus, as pointed out in Chapter 4 (Section 4.3) Smolensky (1988), maintains that subsymbolic models accurately reflect the microstructure of cognitive processes, arguing that classical, symbolic models only provide an approximate description at a macrostructure level.

In formulating such a challenge to classical cognitive science, one expects ANN systems to provide empirically accurate models of human cognition. However, some prominent connectionist models have been criticised because they have not faithfully reproduced some of the behaviour and phenomena documented in empirical studies. Thus Pinker and Prince (1988) criticised Rumelhart and McClelland's (1986) verb transformation network and Besner, Twilley, McCann and Seergobin (1990) critically examined the predictions of the grapheme-to-phoneme mappings performed by Seidenberg and McClelland's (1989) network. The general theme of these critiques is that ANN models do not capture all the empirical data pertinent to an understanding of the psychological phenomena that are being modelled. Probably the best tactic for connectionist researchers when they respond to such criticisms is to tone down their claims about the models. They should point out that their simulations play the same role in cognitive research as animal models play in medicine and biology. Biological models are nowadays explored to shed light on a variety of mammalian functions. Thus, the simple nervous system of the mollusk Aplysia is studied in the hope that its neural functioning may elucidate the neural mechanisms of plasticity underlying the learning of more complex organisms. In this sense, a biological model is a simple, conveniently studied living system that is analogous to a more complex biological system - such as the human brain - in certain important respects relevant to the issues being explored.

Other models are mathematical or conceptual. These abstract, mathematical models are characterised by sets of equations that are useful in describing and predicting the behaviour of the system. Virtually all areas of quantitative neurobiology apply mathematical models in some way. Likewise in the cognitive and brain sciences there is a long tradition of mathematical modelling. ANNs constitute one type of
mathematical approach in the cognitive and brain sciences. However, because they contain nonlinear properties which render a purely analytical approach problematic, ANNs are typically simulated as computational systems.

In both biology and cognitive science models are metaphors, and they are only as useful as their resemblance to the actual system that they represent. In some cases animal models are very useful for understanding human biology, because they have close anatomical and physiological similarities to humans. The visual cortex of monkeys is very similar to that of humans, and has been used extensively to study the functioning of the visual system in both human and non-human primates (e.g. Thompson, 2000, p. 261). However, in some other respects they may be poor models. For example, monkeys hardly constitute good experimental models for investigating human speech processing mechanisms! The same qualification holds true about the use of computer simulations as experimental tools.

ANN models are sometimes difficult to interpret because it is often unclear how they accomplish the tasks that they have been trained to do. "One thing that connectionist networks have in common with brains is that if you open them up and peer inside, all you can see is a big pile of goo" (McMillard, Mozer & Smolensky, 1992, p. 970). There are a number of reasons why ANNs are difficult to understand. First, they are rarely completely predetermined by the modeller, who instead merely specifies the general topology of the network, and makes use of a generic learning rule such as backpropagation to 'grow' the relevant structures in the network. Thus, one does not need a theoretical account of a to-be-learned task before a network is created to do it. Second, general learning procedures can train networks that are extremely large so that their size and complexity makes them difficult to interpret. For example, the network designed by Plaut and Shallice (1993) to compute a mapping from graphemic to semantic and phonemic representations contained several hundred units. It is far from straightforward how such a large network learns to perform the mapping in question. Third, most interesting ANNs incorporate nonlinear activation functions. The nonlinear attributes of these models render them more powerful than networks containing only linear activation functions (e.g. Jordan, 1986), but it also means that descriptions of their behaviour will necessarily be difficult. In some cases only a qualitative (rather than a precise quantitative) account of the behaviour of these nonlinear systems can be given (see e.g., Elman et al., 1996).

As networks grow in their complexity it becomes increasingly difficult to analyse the similarity structure of the hidden units by direct inspection. Because of the difficulties associated with explaining the behaviour of the networks, various interpretive strategies have been proposed by researchers. Some researchers have tried to study how individual processors contribute to a networks' performance by testing the effect that removal of a particular unit has on a network's behaviour. Another common second strategy is to perform statistical analyses of the connection weights in trained networks. Hanson and Burr (1990) discuss several techniques that have been used to explore network structure, including
compiling frequency distributions of connection strengths, computing descriptive statistics to get an idea of global patterns of connectivity in networks, and performing cluster or principal component analyses of the weight matrices in trained networks. Thus, Small et al. (1995) used principal component analysis to analyse correlations (i.e. similarities) among the activity patterns of the units in the network’s hidden layer. Another strategy is to map out the response characteristics of each processor in the network. Moorhead, Haig and Clement (1989) employed the generalised delta rule to train an ANN to identify the orientation of line segments presented to an array of input units. They tried to establish whether the hidden units in this system developed centre-surround receptive fields similar to those found in the lateral geniculate nucleus of the visual cortex. They stimulated each input element individually, and then plotted the resulting activations in the hidden units.

The diversity of these methods shows that providing explanations of ANNs is a nontrivial task. Hecht-Nielsen (1990, p.10) makes this point in a hyperbolic form:

There is a growing suspicion that discovering [how a network does its job] may require an intellectual revolution in information processing as profound as that in physics brought about by the Copenhagen interpretation of quantum mechanics.

Nevertheless, the prognosis for evaluating and analysing ANN models is not as bleak as some critics would lead us to believe. Oussar and Dreyfus (2001) have recently argued that ANNs should be viewed as grey, rather than black boxes. Although these networks are complex nonlinear systems, and their behaviour difficult to analyse, they are not completely opaque to researchers. For example one of the main advantages of ANNs for cognitive neuroscience and neuropsychology is that they can be used for doing lesioning experiments to establish what effect damage to a particular portion of a network will have on its performance. Classical cognitive science researchers typically pursue a top-down methodological strategy, aiming to design a model accounting for the input-output mappings associated with a particular cognitive process using a set of symbolic representations an algorithmic, rule-based descriptions of the operations that must be performed on these symbols to generate appropriate output. The methodology used by connectionists to explore cognitive phenomena, and to develop explanatory accounts of aspects of cognition, is very different. As Clark (1990, p. 299) has observed, the connectionist performs a kind of Copernican revolution in cognitive science:

"...the connectionist effectively inverts the usual temporal and methodological order of explanation, much as Copernicus inverted the usual astronomical model of the day by having the earth revolve around the sun instead of the other way round. Likewise, in connectionist theorizing, the high-level understanding will be made to revolve around a working [connectionist network] which has learnt how to negotiate some cognitive terrain".
That is, rather than formulating a computational theory of a particular input-output mapping associated with some cognitive capacity, the ANN modeller tries to construct a connectionist network capable of actually performing the input/output mappings in question. This may be a difficult task, requiring experimentation with a number of different network configurations and learning rules. It is only when such a working network has been created, and after the researcher had performed a range of quantitative analyses of the network's performance, that a computational theory of the capacity in question can be articulated. In terms of Marr's (1982) conception of levels (which is described in Section 4.3), connectionists follow a bottom-up methodological strategy; they construct algorithmic (i.e. level-2) explanations of cognitive phenomena on the foundations of working models (i.e., 'level-3' implementations).

Because ANN researchers often pursue such a bottom-up approach, their methodology can be opposed to classical explanation strategies. The goal of ANN modelling is to construct explanatory theories that use a small number of principles to account for a broad range of cognitive phenomena. These principles are independently motivated in the sense that they are not task or phenomenon-specific, and they are based on first principles because they are derived from neural processing mechanisms. If the principles are identified correctly, modelling should merely incorporate domain-specific variables such as a content data set, a training regime, and an appropriately configured network structure. The relevant generalisations about the domain in question should then fall out of the model. Connectionist don't take complex cognitive phenomena and dissect them into smaller components as classical researchers do. They start with a set of components (the principles) and attempt to construct simulations of complex cognitive phenomena using these components as their only tools. Their activity is synthetic rather than analytic. Properties that define the particular architecture that a connectionist is using, such as the network topology, learning procedure, learning rate, and activation, impose constraints on the processing and learning behaviour of the net. The main question of interest for connectionist modellers is to establish what kinds of interesting phenomena can be explored and simulated using these building blocks. This synthetic strategy requires one to start the investigation by proposing a particular architectural description, and then analysing the architecture and the model's performance in order to show how it contributed to our understanding of the phenomenon (see also Seidenberg, 1992 in this regard).

An interesting implication of this bottom-up methodology is that connectionist theories of cognitive phenomena are only as good as the network implementations from which they have been derived. To the extent that these implementations incorporate features that are neurophysiologically and neuroanatomically unrealistic, the cognitive theories they license will be suspect. However, as already suggested in Section 6.2, connectionists who seek to develop theories of human cognition do not restrict themselves to concepts such as 'units' and 'connections', but focus on activations patterns and employ terms borrowed from dynamical systems theory to describe the coupling between mind and the world,
and the temporal aspects of cognition. They regard their networks as theoretical tools and would argue that constructing simulations requires one to work at a greater level of detail than that entailed by 'mere' verbal descriptions. Designing a simulation is not just an automatic process of running a network, but requires the researcher to make important decisions about the architecture of the ANN model, as well as the training data, training parameters, and testing of the trained network, and these decisions are typically derived from theory. In this regard the use ANN simulations in cognitive research resembles the way in which quantitative models are routinely employed in mathematical psychology and economics.

Moreover, there is no reason why ANN modellers should not simply pursue implementation models. Part of the problem here is that we are still ignorant about what exactly the computationally significant properties of biological neural networks are. Indeed, as others have pointed out (e.g., Churchland & Sejnowski, 1994) what is required here is a "co-evolutionary research strategy"; a strategy in which connectionist researchers and neuroscientists guide each other to the features of neurons and their synaptic connectivity that are essential to their cognitive function. In this regard, ANN modellers should, of course, be willing to constrain their processing units, connections and learning rules in accordance with the latest work in neuroscience. But equally, neuroscientists can also gain valuable insights from the speculations that connectionist researchers have presented about mechanisms that may play a role in biological brains.

6.4 THE RELEVANCE OF CONNECTIONIST MODELS TO COGNITIVE SCIENCE

The main theme of this thesis was to consider the relevance of ANN modelling to cognitive science, and it is therefore appropriate to tease out some of the main threads emerging from the discussion. A few general problems associated with the classical, information processing approach to cognition were mentioned in Chapter 3. The symbolic paradigm fostered a very active and coherent research programme, but has not been an unmitigated success. The main difficulty confronting proponents of the classical, computational conception of mind is to reconcile the two subtheories associated with the perspective; the computational and the representational components. The computational conception yields a very powerful language for describing cognitive phenomena as algorithmic processes. But computation is essentially a syntactic process involving the formal manipulation of symbolic tokens, and the problem is to explain how these symbolic objects acquire their semantic content in the first place. In short, the classical view is in need of an ontology for the symbolic tokens that play such a dominant role in the theory. Note that it is one thing to simply assert that human cognition involves a system of representational structures, but quite another to show how these structures secure their representational content, that is, their ontology.
The representational issue has proven to be the Achilles heel of the classical view of computation, and several researchers have based their critique of the classical computational view on what Bickard and Terveen (1995) call "encodingism", a presupposition that the mind consists of a system of encoded symbols without a clear explanation of how the encoding process occurs and how the symbol actually manages to represent the entity or state of affairs it denotes. Some researchers have even despairied of finding a solution (Lakoff, 1987). As pointed out in Chapter 4, the PDP variety of connectionism rests on a different concept of representation in terms of which representation does not reside in individual tokens but emerges from the collective activity of a potentially large number of units acting in concert. Some researchers have suggested that these subsymbolic connectionist representations can help to resolve some of the difficulties that cognitive researchers confront in "grounding" symbols to the world. A famous example to show why grounding symbols is a problem for classical theories is presented in Searle's Chinese room problem. One possible reason why the connectionist paradigm may be relevant to cognitive researcher, is because it fosters a fresh perspective on some of the problems associated with the 'symbol-processing' assumption underlying the classical approach.

6.4.1 Using a subsymbolic approach to break out of the Chinese room

The crucial issue for classical cognitive science is to explain how representations acquire their meanings. How do we relate the representational structures in a computational system to entities and situations in the world? Searle's Chinese room argument serves to accentuate this problem, because it stresses the intentional property of human cognition. Searle (1980) contends that our mental states and mental representations are about things, but mere symbols have no intentional content by themselves, they need a process which relates them to the world. He presents the following argument. Assume the dreams of AI researchers are realised and they succeed in developing an accurate model of the mind, which passes the Turing test (i.e. it has such humanlike communication skills that it is capable of fooling an independent observer into thinking it is human). Researchers adopting a strong version of the computational thesis maintain that such a program understands language in the same way that a human does. Searle (1980) rejected this conclusion, and presents a somewhat infamous argument (because of the amount of criticism it elicited from other researchers) in which he sketches a thought-experiment intended to show that even if by some dramatic breakthrough in AI technology a computer is developed that is capable of using language, it still would not exhibit any real understanding of the words and sentences it utters, but would simply be mindlessly following instructions.

To prove his point, Searle asks us to imagine the operation of such a program by analogy with a human who speaks no Chinese but has learned to follow a set of instructions (formulated in English) which he steps through by hand. The human is isolated in a secure room, and batches of Chinese writing are
passed into the room. Although the human understands no Chinese, he reads the instructions given to him in English and uses these to perform a series of manipulations involving symbols on paper. These symbols are in fact the Chinese symbols that represent the output of the program. Every time the human inside the room is given a sentence in Chinese, he consults his instructions and replaces the squiggles with other squiggles and passes out his response to the people outside. Nobody outside the room is aware of what is going on inside and just witness the room as somehow providing appropriate responses to their queries. Searle draws two conclusions from this thought experiment. He argues firstly that it is clear that the human does not understand any Chinese, and is just manipulating what to him are meaningless squiggles. He contends secondly, that the scenario inside the room captures the symbol-processing behaviour of digital computers, and that just as much as the man in the Chinese Room does not understand any Chinese, so too does a computer program not evince any understanding of the symbols that it processes. Many problems have been raised in connection with Searle’s argument. The most common retort is that whereas it is intuitively clear that the human does not understand Chinese, it is quite plausible to regard the system as a whole (i.e. the human, room, pens, paper, and instructions taken together) as showing understanding. Searle dismisses this idea maintaining that it is ridiculous that a system of pens, paper, and the human could ever understand anything, but this may be a limit of his intuition.

The point about Searle’s Chinese room argument is that it highlights the need for classical cognitive science to explain how representations acquire their meanings. In virtue of what does a symbol, which in a computer is just a formal entity, acquire its signification, and represent what it purports to represent. Searle’s main contention is that our mental states and mental representations are about something, but mere symbols have no intentional content by themselves, they need a process which relates them to, or ‘grounds’ in, to the world out there. Connectionist models suggest at least one solution to this symbol grounding problem, because in an ANN, symbols are emergent from the subsymbolic activities of the connectionist processing system. The symbolic representations emerge from low-level statistical operations occurring in direct interaction between an agent and the world. The basic assumption underlying connectionist approaches to meaning is that concepts, are not formed in isolation from the world, in abstraction and in some completely objective manner. Instead they are constructed in relation to the life-world of agents, through a perceptual/motor apparatus, and they are linked to goals, needs, and actions. Hence, the primacy of direct, unmediated interaction between agents and the world, and the embodied nature of cognition is emphasised (see Barsalou, 1999; Clancey, 1999).

It is for this reason that some proponents of the connectionist approach, such as Chalmers (1993), have argued that the differences between symbolic and subsymbolic computationalism frees ANNs from the criticism associated with the Chinese room. Chalmers (1993, p.3) believes that Searle’s assumption that syntax is not sufficient for semantics (which the Chinese room argument is suppose to prove) is simply
false, because it eschews the physical aspects of the brain as a processing mechanism.

On the highest level, the human mind seems extremely flexible, producing the very antithesis of rule-following behavior; yet at the bottom level, it is made up of a physical substrate, consisting of such entities as elementary particles and electric charges, whose actions are determined by the laws of physics.

In other words, the human brain follows the rules of physics which are merely syntactic rules. Since the brain has internal meaning (if anything does), then the axiom that syntax is not sufficient for semantics is false. In the case of the human brain, it clearly is possible. For example, one implication of Searle’s position is that if the argument is valid, it can be taken to its logical extremity as follows: Individual nerve cells in the human brain do not process symbols. They process pulse sequences with varying frequencies. Therefore, the human brain does not really understand language - it merely processes pulse sequences! Since ANNs model the operation of the brain at a functional level, the force of the Chinese Room argument is weaker when directed at them than when it is directed at the symbolic paradigm of mainstream computational cognitive science.

Moving beyond representationalism

Classical researchers proceed from a syntactic picture of the brain, which involves depicting all or most of human cognition as involving something like logical operations applied to something like linguistic (sentential) structures. In contrast, a typical PDP connectionist network does not employ grammatical strings nor, *a fortiori*, processing operations sensitive to the structure of such strings. Instead, they make use of a prototype-based representational in which closely related data items become encoded in neighbouring regions of weight space. It is because networks exploit similarity in this way as a *semantic metric* that they can generalise and respond appropriately when presented with incomplete or noisy input data. However, it is not clear that their distributed representational scheme amount to the provision of a genuine syntax, because it does not allow for the representation of complex sentential information in a straightforward manner. As Fodor and Pylyshyn (1988) point out, there is no analogue in distributed representations to the logical operations of removing an element from one string (complex representation) and adding it to another.

Ramsey, Stitch and Garson (1990) describe a thought experiment in which a simple feedforward, multilayered ANN, using distributed representations, is trained to evaluate the truth of propositions.
During training propositions are presented at the input units as patterns of binary activation, with their truth values indicated. When training is complete the weight matrix is analysed to decipher the relations that map propositions to truth values. Consider now the issue of which one of the propositions in the training set actually contains the assertion Mary loves John. Where in the network is the knowledge associated with the proposition, or any of its constituents encoded? Since distributed representations are used, the answer is, of course, everywhere; the whole weight matrix is responsible for encoding the mapping relationship. Also, how do we determine which part of the network should be cut away in order to force the network to forget only one, and none of the other, propositions? Similarly adding new propositional knowledge to the network involves making adjustments to potentially the whole network. Clearly this representational scheme contrasts sharply with the one used in the classical approach. In a classical (symbolic) semantic network, for example, propositions can be deleted by removing arcs and nodes, and propositions can be added in a corresponding way.

Because there is such a crisp distinction between the distributed representational systems used in ANNs, and the classical approaches to representation, some researchers criticise ANN modellers for retaining an unnecessary burden of classical cognitive science; that is, a commitment to representations and representational talk (van Gelder, 1998). Successes in the development of robot systems that do not employ internal representations have inspired some theorists to argue that representation is not at all necessary for cognition. In the field of situated robotics researchers have developed small robotic systems capable of navigating an environment, and of displaying simple, intelligent ecologically relevant behaviours such as the ability to avoid obstacles, move over terrain and search for particular types of objects or stimuli (Brooks, 1991). The assumption here is that intelligence can only be readily described in the context of a cognitive agent that is connected to its environment via sensory input and motor effector output. Intelligence is an emergent property of the intimate relationship between the agent and the environment rather than something intrinsic to the agent itself. In pursuing this line of research, cognitive scientist and AI specialists have abandoned any attempt to build a representational model of the world and rely on direct processing of the environmental stimuli.

Although this framework circumvents many of the problems that are encountered in the process of building complex representations, it appears to echo some of the claims of the behaviourists in their emphasis on the study of behaviour and leaving the mind as a ‘black box’. Moreover, the successes achieved in situated robotics are mostly limited to perceptual and motor applications so it is not clear whether the same approach can be used to model and account for aspects of higher cognition. Still, the ubiquity of representational talk in brain theory and ANN modelling deserves scrutiny. It is likely that some kind of internal representation will be useful in constructing and discussing connectionist models, but precisely how much and what kind is a matter deserving serious concern. It is in this context that ANN modelling presents an alternative to either the strong (symbolic) representational assumptions
underlying the classical approach, and the non-representational stance adopted by researchers in situated robotics. Connectionist researchers acknowledge that representational structures may be needed to account for aspects of higher-cognitive processes, but their commitment to the representational position is much weaker than that of classical researchers.

6.4.3 ANNs as dynamical systems: A non-classical perspective on cognitive functions

One way in which ANN models seem to go beyond the classical representationalist thesis, is in their close association to the dynamical systems perspective, which has been gaining ground in the cognitive sciences. The notion of a 'dynamical system' is relevant to a broad range of mathematical and scientific contexts, and as a result the term is used in different ways. Dynamical approaches to cognition can be traced back to at least the cybernetic era in the 1940s when researchers tried to discover the brain’s functioning by integrating aspects of information theory, dynamics and computation. These approaches were not intensively pursued during the 1960s and 1970s when the computational approach flourished under pioneering work by researchers such as Bruner, Goodnow and Austin (1952), Miller (1956), Newell and Simon (1972), and even Chomsky (1965) whose work was at least broadly consistent with the computational thesis. More recently the idea that dynamics is a relevant framework for understanding cognition has resurfaced under the instigation of developmental researchers such as van Geert (1993) who applied the framework to language acquisition, and Thelen and Smith (1994) who used concepts from dynamical systems theory to elucidate aspects of cognitive development. For instance, Thelen and Smith (1994) described the development of kicking and reaching in infants in terms of dynamical notions such as the stability of attractors in phase space defined by body and environmental variables. They also contend that higher cognition is ultimately rooted in the types of spatial skills learned in infancy (an idea echoed in cognitive linguistics) and that higher cognition will thus also best be understood dynamically. Their approach contrasts with traditional information processing theories of development in which new developmental stages are caused by brain maturation and the increasing ability of infants to reason logically.

Van Gelder (1998), for example, denies that psychological processes are computational. He argues that cognitive systems are dynamic, and that cognitive states are not relations to mental symbols, but quantifiable states of a complex system consisting of (in the case of human beings) a nervous system, a body and the environment in which they are embedded. Cognitive processes are not rule-governed sequences of discrete symbolic states, but continuous, evolving total states of dynamical systems determined by continuous, simultaneous and mutually determining states of the system’s components. Representation in a dynamical system is essentially information-theoretic, but the vehicles for encoding
and transmitting information are not symbols, but state variables or parameters. For this reason some researchers such as Thelen and Smith (1994) contend that dynamical systems theory provides a more appropriate language for describing cognition than the traditional information processing notion in terms of which cognitive operations are construed as transformations of a computational system moving from one static symbolic structure to the next. In terms of the dynamical system conception, cognition can be characterised as a continual coupling among brain, world and mind, that unfolds in real time, as opposed to the discrete time steps of digital computation. The emphasis of the approach is on how the brain, body, and environment as a whole changes in real time, and dynamics is proposed as the best framework for capturing that change. This is said to be opposed to the classical computational perspective's focus on internal structure (i.e. a concern with the static organisation of information processing and representational structures in a cognitive system). This emphasis on change and movement is the distinctive feature of dynamical approaches.

Computational theories in cognitive science are theories of structure, and make claims about information processing and the functional structure of mental states. These theories typically assume that information processing involves the manipulation of explicit, static structures, and often eschew issues relating to change (e.g. the problem of explaining how high level symbols may emerge from a lower substrate). Dynamical approaches contribute a much needed characterisation of continual change, which yields a framework for explaining complex relations between the brain, body and the environment. Computational approaches contribute notions of mechanism and equivalence classes of mechanism that help to clarify the functional and adaptive behaviour of complex systems. Some researchers attempt to separate dynamics and computation, but the arguments put forward in support of such a position are somewhat problematic. For some complex adaptive systems a full understanding will require a rapprochement between 'computation talk' and 'dynamics talk'. Connectionism is at least one approach that helps to make headway on this task.

ANNs are clearly dynamical systems in that they contain variables for the activation of the various units and for the connection weights linking these units, and they also contain nonlinear equations for updating the activations and changing weights. Moreover, the complex nonlinear behaviour of recurrent networks (which instantiate dynamical systems) is often best analysed with the aid of concepts borrowed from dynamical systems theory, relating to the temporal aspects of the activation patterns that networks manifests as they learn and execute processing tasks. In some respects this dynamical perspective on the functioning of ANN models provides a close fit to neuroscience because it highlights the dynamics of activation flow in complex nonlinear networks (of processing units). Thus, Churchland (1989, p. 280) remarks that the goal of connectionist researchers is to ground theories of psychological and brain functions in actual neuro-physiological mechanisms, suggesting that:
"[The] brain represents various aspects of reality by a position in a suitable state space, and the brain performs computations on such representations by means of general coordinate representations from one state space to another."

He invites one to view concepts as points in a partial state space of a dynamical system. State-space is the vector space where different activation patterns of a network can be placed, by considering each state of the system as a vector. According to this view, a distributed representation will be a point in such a space (with as many dimensions as neurons), and conceptual representations can be investigated in terms of their positions and trajectories in state space using data analysis techniques such as hierarchical clustering, principal component analysis, or discriminant analysis. By using such descriptions, distributed representations offer functionalities similar to classical symbols, but they are obtained in a rather different manner. Like symbols, each distributed representation has a unique form (an activation pattern), which may refer to a distinct object in the semantic domain. However, the state-space of ANNs, which could be discrete or continuous, is generally much "denser than the sparse discrete space" of symbolic representations, and unlike symbolic representations, the form of distributed representations is not completely arbitrary, because similar objects will be in close related representations in state-space (Churchland, 1989, pp 280-282).

The main problem, however, is to elaborate the dynamical systems approach into a coherent approach to cognitive explanation. It is difficult to see how, in its current state, it can be applied to yield detailed, testable theories of cognitive phenomena such as language understanding and reasoning (Townsend, 1992). The terminology associated with dynamical systems may be suitable for capturing the overall trends in the behaviour of a system, but not for yielding insight into the fine-grained aspect required for an elucidation of cognitive mechanisms. In principle, dynamical models could be supplemented with representational resources in order to achieve more revealing explanations. For instance, it is possible to treat certain parameter settings as inputs, and the resultant attractor as an output, each carrying some representational content. Furthermore, dynamical systems theory easily handles cases where the 'output' of a system is not a single static state (the result of a computation), but is rather a trajectory or limit cycle as exemplified by the ANN simulations of reading and reading disorders (e.g. Hinton & Shallice, 1991). This approach embraces both the representational characterisation of internal cognitive processes, and a dynamical system's characterisation of the brain's overall function. For this reason, Horgan and Tienson (1994) suggest that ANN modelling yields a non-classical conception of cognition, a theoretical framework that makes it possible to explore the dynamics of cognitive processes, and to integrate a dynamical systems and a computational perspective on cognitive functioning.
6.4.4 Reconciling the connectionist and symbolic accounts of cognition

As a final point to consider, it is worth noting that ANN models can be shown to be relevant to cognitive researchers without arguing that they should replace classical models. Many researchers are critical of a whole-sale acceptance of connectionist models in cognitive science. Norman (1986) suggested in his assessment of ANNs that they appear decidedly "low-level" entities and that it is unclear of what value these networks are in the typically high level cognitive phenomena traditionally studied in cognitive science. Likewise, Pinker (1997, p. 112) refers to the ANN modelling approach to cognition as "connectoplasm" suggesting that they may be appropriate for modelling aspects of cognition relating to processes that occur without much conscious effort such as perception and attention, but that they do not offer a suitable theoretical vocabulary for probing higher cognitive functions. As noted in Chapter 4, Smolensky (1988) offers a rather different assessment, arguing that connectionist descriptions may offer more precise and accurate, albeit very fine-grained, explanations of cognitive phenomena. According to this interpretation, classical theories are only approximations of statistical and subsymbolic processes that are more accurately captured by connectionist descriptions. In view of these different conceptions about the contribution that ANN models bring to cognition, it is safe to say that their status in relation to classical models remains somewhat controversial. Some of the research surveyed in the previous chapter suggests that coping with the constituent structure of sentences is a difficult (but not impossible!) task for ANN models. It is this constituent structure that underlies what Fodor and Pylyshyn (1988) refer to as the mind's systematicity. If it is accepted that ANN models are less suitable than classical models for modelling these types of structures, the mind cannot be a connectionist network. Cognitive explanations must use the language of symbol systems, and connectionism is confined to accounting for the implementing of the system.

Clark (1990) presents a conciliatory position in which he argues that connectionist and classical approaches need not be regarded as completely opposed to one another. He makes the reasonable point that the computational architecture of the brain is not necessarily uniform, and may support a variety of virtual machines: symbolic, connectionist, and other as yet undiscovered types. The behaviours that pose the greatest challenge to connectionism, may well depend on a non-connectionist virtual architecture. Clark speculates that for some evolutionary recent behaviours such as logical inference, the syntactic aspect of language, and mathematical reasoning, the mind may simulate a classical von Neumann architecture. He suggests that a von Neumann architecture may have emerged from the acquisition of mechanisms for representing possible states of the world using external artefacts.

In the case of multiplication, for example, learning may have resulted from pattern-matching skills that allow one to recognise that the product of 7 and 8 is 56. It is not too difficult to construct an ANN model
to learn to approximate this type of pattern recognition task, but much more complex to get it to produce the correct, precise answer, as a study by Anderson (1998) showed. More difficult cases such as 28795 multiplied by 6678123 are difficult for most of us to do in our heads. One possibility is that these two tasks are served by different cognitive capacities, and that the mind has a hybrid rather than a uniform architecture. The case of patient P.S., discussed in Sokol, McCloskey, Cohen, and Aliminoza (1991), lends some support to these speculations. She was a 38-year-old financial analyst when a malformation of blood arteries and vessels in her brain haemorrhaged, causing damage in the temporal lobe of her left hemisphere. Extensive psychological testing revealed a startling disruption in her ability to do simple arithmetic calculations. She made a substantial number of errors with simple arithmetic problems like 6×7 and 8×9 getting only 20 percent of them right. Even worse, with single digit calculations involving a zero she was always wrong. She would say that 4×0 is 4. But she made only two errors involving the application of rules in 66 long multiplication problems (not containing zero's!) such as 187×29. This case leads us to question our ordinary assumption that arithmetic and math form one single, indivisible collection of information. Instead, the odd patterns in the disruption of P.S.'s arithmetic abilities would seem to suggest that simple fact retrieval and rule knowledge are distinct cognitive abilities, served by different brain structures.

In long multiplication tasks we might make use of a pen and paper, and arrange the problems so as to be able to apply the rules of long multiplication. Using these external artefacts the problem can be neatly composed into a series of simple pattern-matching stages with the result of the intermediary stages represented on paper. Although this is undeniably a symbol manipulation task, the symbols are in the world, not the head. It is nevertheless possible to learn to do long multiplication in the head. According to Rumelhart et al. (1986) we achieve this by building a mental model of the in-the-world symbol manipulations involved in the multiplication tasks. On this account, symbolic representations in the world form the basis for the development of mental symbolic representations. The foregoing implies that the behaviours in question are best explained by the operations of a von Neumann virtual machine implemented on a connectionist machine. The mind is not uniform, it is hybrid. For some behaviours involving memory and perception the appropriate cognitive models will be connectionist. For other behaviours, such as mental arithmetic and logic, the appropriate models will be variants of physical symbol systems. The evolutionary development of biological organisms seldom proceeds in a neat and premeditated way, but often relies on makeshift solutions, and tinkering with the material available. For example, Davis (1998) presents some interesting evidence suggesting that the evolution of language may have stemmed from purely accidental aspects of hemispheric specialisation associated with hunting tasks (e.g. disk throwing), and the memory requirements imposed by food storage during averse weather changes. There is no clear evidence at all for any systematic evolutionary programme leading to the development of higher cognitive abilities, and so the assumption of a principled ordering of levels of analysis may be problematic in the case of human cognition.
Similar ideas are presented by Marr (1977) who suggests that there may be two main types of cognitive problems in the cognitive sciences, and that these require different explanation strategies. He suggests that a science of intelligence requires either a "Type-1" exploration based on theoretical understanding of fundamental (axiomatic) task constraints, or "Type-2: implementations of intelligent performance effected by the simultaneous action of a considerable number of processes "whose interaction is its own simplest description" (Marr, 1977, p. 41). Examples of Type 2 problems are general reasoning and language understanding, both of which illustrate an interaction between many variables. The implication that can be teased from Marr's analysis is that some cognitive problems can be attacked using classical style decompositional (e.g. divide-and-conquer) strategies, but Type 2 problems may not be amenable to such approaches. On the other hand, at least in theory ANNs may prove useful in the case of Type 2 problems because connectionist researchers do not seek an axiomatic task analysis before starting to model the processes in question, but instead build a model only loosely specified by considerations of abstract competence (i.e. what functions the system should be able to perform). These systems may therefore be more viable than classical approaches in the case of Type 2 problems.

Although the connectionist framework diverges from the classical approach in that it adopts a different level of theorising, it does not entail a complete rejection of the classical, symbolic approach. Even Smolensky (1988, pp. 12-16) concedes that core aspects of some cognitive tasks relating to planning and problem solving may require explanations using classical terms. However, he rejects the position that these processes are produced by classical algorithms. According to him, cognition is at the fundamental level controlled by a subsymbolic, connectionist processor.

6.5 A FINAL RECKONING

There is little doubt that connectionist models are a force to be reckoned with in the cognitive sciences, even if the approach is marked by a number of underlying philosophical disputes and unresolved theoretical issues. Connectionism has been adopted by many researchers in the cognitive sciences, it is now a productive area of research that generates new models and new research questions. Yet in many ways, it is a research programme that is still inchoate, because new learning algorithms and areas of application are presented at a fairly regular rate, and these often push the programme into new directions. On the other hand, it is now more than a decade since the resurgence of the connectionist movement in psychology and the cognitive sciences, yet it shows little sign of abating. In fact, ANN models have been adopted as both a technological toolkit and a source of theoretical inspiration by cognitive researchers.
6.5.2 The future of connectionism: where is it heading?

It is still much too early to say something definite about the contributions that connectionist modelling will ultimately make to cognitive research, but at the current state of development it has been mostly useful in highlighting some stumbling blocks for the unfolding scientific understanding of the mind. Three of these are mentioned below.

Researchers need to pay more attention to the role that innate structures play in cognitive learning and processing. Many of the interesting ANN models that have been developed to simulate higher cognitive functions such as language understanding do not learn everything from scratch, but have a predetermined architecture, that is specified by the modeller. It is probably somewhat naïve to expect a network to be able to inductively discover the extremely complex functions characterising such aspects of cognition. It is for this reason that there has been a tendency among researchers in AI, particularly in the field of natural language processing, to experiment with hybrid architectures combining insights from both the connectionist and the classical paradigm. One possibility suggested by Miikkulainen and Dyer (1991) is to use ANN models for constructing semantic representations (by exposing them to relevant content data) and to make use of symbolic rule-based systems for some syntactic aspects of sentence processing. The use of such constrained architectures and hybrid systems may become more common in this field.

An important challenge that has been issued to connectionism stems from the work of researchers adopting a dynamical system’s perspective (van Gelder, 1998). As explained earlier in this chapter they regard temporal aspects as a central theme in cognitive modelling because cognitive systems are embedded in the world, and have to carry out their tasks in real time. Because the computational resources of the brain are finite, humans often exploit aspects of the environment (e.g. pencil and paper) as an extension of the cognitive system. This ‘situatedness’ of cognitive systems needs to be reckoned with if we are to develop an accurate picture of the type of computational tasks that neural systems perform. Thus, while looking inside to the brain and the results of neuroscience, one should not ignore constraints and resources imposed by the outside, that is, the physical context in which cognition is carried out (Clark, 1995).

The problem of scaling up has not yet been adequately tackled by ANN researchers. Minsky and Papert (1988, p. 180) argue that the entire connectionist enterprise may be built on “quicksand”, because:

"...it is all based on toy-sized problems with no theoretical analysis to show that the performance will be maintained when the models are scaled up to realistic size. The
connectionist authors fail to read our work as a warning that networks, like brute force, scale very badly".

Most of the ANNs studied today comprise are many orders of magnitude smaller than the processing structures in biological brains. It is not clear that the principles of training and processing which work for these small networks will continue to work as we develop larger, more realistically scaled network models. In pursuing such a task, the data will become much messier than in the past, because scaling up involves learning how to develop mechanisms allowing highly structured neural organisations to deal with a variety of mixed stimuli, and the noisy, temporally patterned aspects of information. There is still a large gap between this goal and our current understanding of the computational mechanisms underlying biological brains.

Will ANN models take over the cognitive arena? One of the main threads running through the preceding chapters is that these models are interesting and innovative, but there is reason to believe that, for some time to come, they will survive next to classical, information processing models of cognition instead of supplanting these. Even von Neumann, who is now generally considered to be one of the foremost developers of serial computational architectures, accepted that brain functioning can be analysed in terms of serial as well as parallel aspects (von Neumann, 1958). The transmission of messages by individual neural units is so slow compared to the elementary units in digital computers that the parallel activity of many neural units is evidently essential for many kinds of computation. However the output mechanisms in computational systems often exhibit serial processing characteristics (e.g. humans can only pronounce one syllable at a time during speech processing). Further, from what we know of the brain it does not appear to have a completely uniform design like a digital computer, but comprises a collection of systems and subsystems which are of different evolutionary ages. For example, neural structures in the brain stem, such as the pons and the medulla, and limbic structures such as the amygdala and septum, predate the development of the neocortex in mammals (Thompson, 2000, pp. 17-18). There is scant reason to expect any single type of processor model to apply uniformly throughout the ensemble of neural systems making up the brain.

6.5.2 Concluding comments

Some cognitive researchers are inclined to regard ANN models as a return to earlier associative approaches to language and reasoning. However, the characterisation of ANN models as just associative processors is somewhat facile because it eschews the fact that multilayer ANNs often include nonlinear mechanisms as well. The ANN models typically used in cognitive science contain hidden layers and
nonlinear activation functions. Connectionist models endowed with recurrent connections instantiate nonlinear dynamical systems, and they are therefore much more complex than the very simple associative mechanisms studied in the earlier development of psychology during the behaviourist era. These nonlinear properties enable the systems to transcend the restrictions of linear input-output mappings characteristic of some the early ANN models such as the perceptron. Thus, while the complex, nonlinear aspects of human learning served as a challenge to old stimulus-response theories of learning (see Punktett & Marchman, 1991), they are less problematic for some of the more modern connectionist architectures.

Because many of the ANN models currently studied in cognitive science (e.g. Plaut & Shallice, 1993) are nonlinear systems, they can behave in unexpected ways, mimicking the nonlinear (e.g. U-shaped) learning curve and sudden moments of insight that often characterise human learning. For example, in trying to achieve stability across a large number of superimposed, distributed patterns, a network may discover a solution that was ‘hidden’ in bits and pieces of the data. That solution may be transformed and generalised across the system as a whole resulting in what may be viewed as a qualitative shift. Such a qualitative shift occurred when Elman’s partially recurrent network appeared to ‘discover’ the notion of a word unit (see Section 5.3.2). In this sense new structures can emerge between the specification of an initial architecture, and subsequent exposure to relevant data in a particular content area associated with cognition. It is for this reason that Elman et al. (1996) have argued that ANN models can be used to explore the notion of change, and thus to investigate how intelligent processes may emerge from an initially specified cognitive architecture which is endowed with learning functions. On the downside, even though nonlinear dynamics is an important attribute of ANN models, and one that is of particular interest in the cognitive domain (see Heath, 2000), it also makes them rather complex and unpredictable systems. At this stage, we do not really adequately understand the limits and capabilities of nonlinear dynamical systems so that no final pronouncement about their value as models of cognitive phenomena can be made. Also, there is still controversy even about the defining properties of the dynamical approach in cognitive science because it overlaps in many respects with the computational perspective as several commentators on van Gelder’s (1998) article have pointed out.

One of the main allures of connectionist models is that they implement a form of neurocomputation and they are therefore, at least in principle, able to capture some aspects of the brain’s processing mechanisms. However, even though neural plausibility is a key feature of the connectionist research programme, some issues regarding the biological properties of these networks have not yet been settled, such as:

- Are ANN models closer to mirroring the workings of the brain than their classical, information
processing counterparts? In Chapter 4 it was argued that distributed representations are more consistent with brain and memory functioning, but it is much less obvious that other aspects of ANN models, such as some learning procedures (e.g. backpropagation) used to train them are biologically realistic. If ANNs are neurally plausible, they only capture the functioning of the brain at a rather abstract, functional level.

What aspects of connectionist models are responsible for their empirical successes in modelling psychological functions? Here we are still pretty much in the dark. It may be that the soft computing, 'constraint-satisfaction' processing mechanisms underlying ANNs yield a more appropriate language for describing cognition as Smolensky (1988) has suggested, but it is not clear that these aspects can satisfactorily model higher-level cognitive functions such as language and reasoning, although some programmatic studies (reviewed in Chapter 5) are underway.

At best there are some arguments why connectionist representation are more neurally faithful than classical approaches, because they are based on a computational architecture that captures the operation of the brain on a functional level. In this regard, ANN models seem to carry out a different form of computation than the symbol-manipulation operations characteristic of classical computational architectures. The operation of ANNs are more aptly described using concepts such as the 'spreading of activation patterns' over networks of processing units than in terms of discrete symbolic operations. Of course, this does not imply that the functioning of an ANN cannot be translated into the language of classical computational systems. ANNs are routinely simulated on classical (digital) computational systems suggesting that they have close similarities to classical systems. However, there are significant differences in the theoretical conceptualisation and scientific terms associated with ANNs that sets them apart from the classical computational approach.

Classical researchers emphasise the symbolic nature of human representational structures. Classical systems are governed by rules that describe the conditions under which, and processes in terms of which operations are performed on symbolic representational structures. The rules operate on the representations in a way that is sensitive to their constituent structure. That is, symbolic representations can have a complex structure in that they are composed of symbols that stand in syntactic relations to other symbols. In ANNs the relation between representations and rules is very different because representations are not viewed as formal objects and they are therefore not subject to direct manipulation. The rules in an ANN system operate at the level of individual processing units, specifying how activation is sent to other units and how connection weights are modified. Intelligent behaviour emerges from the way in which interacting units are connected, and not from the direct manipulation of representations. Semantic representations are in a sense ‘grown’ by exposing an ANN model to content data from the
domain of interest. The semantic representations are not just arbitrarily specified by the researcher, but are discovered by the network as it attempts to approximate the input-output mapping in the domain. Moreover, in many ANN models a distributed representational scheme is used in which patterns of activation over a whole network, or a large portion of a network, encode semantic content. Even though both classical and connectionist approaches may therefore be based on the underlying assumption that the brain can be modelled as a computer, they nevertheless yield rather different perspectives on how it represents and processes information. This is true both in so far as the theoretical language used to describe models in the approaches are concerned, and in the type of cognitive architectures that have been proposed in the two approaches. More specifically:

- The ANN approach relies on continuous mathematics rather than the discrete mathematics commonly invoked in classical, symbolic approaches, so that in connectionist approaches mental representations are not conceived of as static symbolic structures resident in memory, but as vectors specifying the activation state of a complex distributed system. Moreover, in ‘attractor’, networks of memory and conceptual structure, mental phenomena are not construed as rule-based, algorithmic processes controlled by a central executive (main program), but as differential equations governing the evolution of a dynamical system.

- Connectionist cognitive architectures are fundamentally two-tiered. The algorithmic specification of the processing mechanisms and the semantic interpretations of the systems' behaviour are located at two different levels, because in order to interpret the semantic content associated with a particular cognitive process, one has to consider the global pattern of activation over many units, and not just the ‘content’ of individual units. The concept of the cognitive unit in a subsymbolic connectionist approach is therefore different from that proposed under the standard symbolic account. In the former such a unit has a specific semantic content, whilst in the latter semantics is only manifest as activations defined over many individual units.

The symbol system perspective underlying the classical approach is inspired by computability theory and the conception of a Turing machine, and therefore yields as powerful a mechanism for dealing with computational functions as can possibly be investigated. However, the assumption about the computational properties of these systems relates to what can theoretically or effectively be computed and does not take practical constraints into account. A TM is not a very efficient processing system and therefore does not necessarily capture all the pertinent properties of a system such as the human brain which has to survive in the real world and therefore needs to cope with environmental challenges in a practical (and not just an effective) manner. Examples such as the ‘frame problem’, which was discussed in Chapter 5, show that a purely effective a solution to some everyday problems give rise to
a combinatorial explosion of different possibilities to explore. Even though such problems can be solved by classical systems, the solutions are not practically feasible because they will take far too long to execute in real time. In Chapter 5 it was also shown that similar problems emerge in regard to classical approaches to commonsense reasoning. Nonmonotonic logics such as Reiter's default logic, McCarthey's (1980) circumscription, and many of the modern variants of these systems, provide interesting alternatives to the strictly monotonic system associated with standard first order logic. These nonmonotonic logics show how defeasible reasoning can be carried out by a logical system, but unfortunately they are subject to complexity issues so that they are not suitable for implementation in practical AI systems. Connectionist models provide an alternative method for dealing with the complexities of commonsense reasoning.

A point that emerges from the discussion in the previous chapters is that the relative advantages of classical, symbolic cognitive science and connectionist approaches are less clear-cut than is sometimes assumed by researchers (e.g. Bechtel & Abrahamsen, 1991). In particular, while ANNs have been used successfully to model aspects of memory and conceptual structure, learning and constraint satisfaction, most ANN researchers have not yet adequately addressed the complex information processing issues routinely tackled in classical approaches to language understanding, reasoning and problem solving. The latter field has contributed more to how natural language discourse is understood, and to how commonsense reasoning is performed. Nevertheless, some connectionist researchers have pointed the way toward ANN systems that may ultimately be able to deal with cognitive processes for which only symbolic approaches have so far been proposed. Moreover, ANN models implement a statistical form of computation that is closer to the computational mechanisms underlying brains than symbolic models. They may therefore serve to revitalise and extend classical approaches by advancing a framework for conceptualising human cognition in terms of neurocomputational principles. Smolensky has even intimated that ANN modelling may ultimately lead us to reconsider our acceptance of the Turing machine as the only viable conception of what computation and cognitive information processing entail. Copeland (2000) suggests likewise that the TM model of computation only accounts for narrow mechanism, and that a broader idea of mechanism may be needed which accepts that the mind may be a machine but "countenances the possibility that information processing machines that cannot be mimicked by a universal Turing machine, and allows in particular that the mind may be such a machine" (Copeland, 2000, p. 9). However, despite the apparent significance of connectionist models for cognitive researchers as a useful supplement to classical approaches, an open question in this area of research is whether it is actually possible to replace classical cognitive science approaches when modelling complex cognitive processes such as language understanding and reasoning. If such connectionist models are developed would they be "eliminative", as Marcus (1998) calls the radical agenda in connectionist research which aims to overthrow the classical approach, or would they simply be implementations of conventional symbolic systems? How can different styles of connectionism (e.g.
eliminative and implementational connectionist systems) be gracefully combined into hybrid systems? These are all still unanswered questions, awaiting further exploration.

The most significant contribution of the connectionist paradigm may be in the new methodology for research that ANN modelling furnishes. Although classical and connectionist approaches ultimately seek to model and explain cognitive information processing, classical theories are more ‘top-down’, relying heavily on computer science principles, whereas ANNs are rather more ‘bottom-up’, aiming to take relevant neurobiological constraints into account. In this respect, ANNs may be somewhat more biologically plausible than the top-down models of classical cognitive science, but whether this is always an advantage remains to be seen. For some projects such as modelling language comprehension not much useful neural detail is available at this stage so that classical approaches still tend to prevail. However, more neural detail is required when one explores cognitive neuropsychological phenomena. Connectionist models have proven useful in the latter context, because they have been employed to perform simulations of neuropsychological syndromes as the Hinton and Shallice (1991) and the Plaut and Shallice (1993) studies demonstrated. In this type of application, ANN models are used to explore the effect of damage on a simulated cognitive system, just as animal models are used in psychology (McCloskey, 1991). Moreover, researchers need not restrict themselves to general speculations about a model’s behaviour, but can actually open it up to study its innards. They can also lesion parts of a network, and study the effects this has on the model’s behaviour. Still, although they can be analysed and studied with lesioning experiments and even though they are not quite the black box technologies of older associative approaches such as behaviourism, they are, in many respects, still ‘grey boxes’ (see Oussar & Dreyfus, 2001).

Even viewed in this somewhat deflationary sense as ‘grey boxes’, the present breadth of ANN models indicates that the approach has considerable potential. Despite attempts to establish a priori limitations on ANN models of higher cognition, probably the only way to determine the ultimate value of the approach is to pursue it with vigour. Perkel (1987, p. 181) remarks that the proponents of any new approach have a tendency “to find areas of application in the spirit of the proverbial small boy with a hammer, who discovers an entire world in need of pounding”. ANN modellers are bound to take up this challenge and to apply their models in new and creative ways. If realistic ANN models of language processing can be provided, then a radical rethinking of some of the assumptions that classical theorists make about the role of symbols and rules in cognition may be needed. It may be that the ultimate description of high and low level cognition resides in the distributed structures of complex connectionist networks, and can only be approximated by symbolic rules. At the very least connectionist research programme has yielded an important set of techniques, based on neurologically plausible principles, that enable researchers to explore basic assumptions about cognitive information processing on a common simulation platform. The future of connectionist cognitive science is therefore likely to have important
implications in either overturning traditional classical approaches, or at the very least in bolstering them with new implementational possibilities.
REFERENCES


191


Besner, D., Twilley, L., McCann, R.S. & Seergobin, K. (1990). On the association between connectionism and data: are a few words necessary? Psychological Review, 97, 432-446.


194


Basil Blackwell.


203


Derivation of the Delta Rule used in the Widrow and Hoff network

We define a measure of the error in the output:

\[ E_p = \frac{1}{2} \sum t_i (y_i - t_i)^2 \]  \hspace{1cm} A.1

In this expression \( t_i \) represents the target output value associated with unit \( i \), and the subscript \( p \) indicates that we are dealing with a specific error pattern. An immediate difficulty is that the error measure, \( E_p \), is calculated for the whole pattern, so that it is not clear what contribution each weight makes to this error. What is needed is to change the weights so as to minimise the partial derivative of \( E \) with respect to each output unit, because this will show how \( E \) is affected by the change in that activation. The partial derivative can be calculated locally, being simply the difference between the desired and actual activations (with the appropriate sign added):

\[ \frac{\partial E}{\partial y_i} = -(t_i - y_i) \]  \hspace{1cm} A.2

This tells us how much the activation of each output unit must be adjusted to reduce the unit’s contribution to \( E \). We can now work backwards because the output unit’s activation can be adjusted by changing the weight of each of its connections with an input unit (bearing in mind of course that the input activations must not be changed!). The problem therefore reduces to computing changes to the weighted connections in the network so as to change \( E \). This entails determining the partial derivative of \( E \) with respect to the weights, which can be achieved by appealing to the chain rule.

\[ \frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial w_{ji}} \]  \hspace{1cm} A.3

The next step is to evaluate the partial derivative of the activation with respect to the weights. The Widrow and Hoff network makes use of a linear activation function, so that we have:

\[ y_j = \sum w_{ji} y_i \]  \hspace{1cm} A.4

and the partial derivative is just the activation of the input unit.
\[
\frac{\partial y_i}{\partial w_{ji}} = y_i
\]  \hspace{1cm} \text{(A.5)}

Hence the partial derivative of the error with respect to each weight (with a negative sign because we are going downhill in the error landscape) can be obtained by multiplying the difference between the target and the actual activation with the activation of the input unit, yielding:

\[
\frac{\partial E}{\partial w_{ji}} = -y_i(t_j - y_j)
\]  \hspace{1cm} \text{(A.6)}

The delta rule now multiplies the negative of this derivative by a learning rate to compute the change in the weight connecting units i and j:

\[
\Delta w_{ji} = \eta y_i(t_j - y_j)
\]  \hspace{1cm} \text{(A.7)}

Substituting \( \delta_j \) for \((t_j - y_j)\) yields

\[
\Delta w_{ji} = \eta y_i \delta_j
\]  \hspace{1cm} \text{(A.8)}
Derivation of the backpropagation rule

The backpropagation rule differs from the delta rule in two important respects. First of all, it is defined for networks that can contain non-linear activation functions. It is not necessary to use a nonlinear activation function, but is often used in practice, especially when the network has to learn a complex mapping. In the training rules specified below, it is necessary for the rule to be differentiable and monotonic. The most common function used is the logistic or sigmoid function which compresses the range of the net input so that the output signal lies between 0 and 1. The sigmoid function is shown below:

\[ a_j = f(\text{Net}_j) = \left(1 + e^{-\theta_j + \theta_j}\right)^{-1} \]  

B.1

The second difference is that backpropagation is typically used for multilayer networks which include a hidden layer in addition to the input and output layers. Recall that the perceptron could only learn linearly separable functions. Multilayer networks trained with backpropagation overcome this shortcoming, and can learn any continuous function. However, the problem with multilayer networks is that it is not obvious from the error computed at the output layer what contribution the hidden layer units have made to the error. The remarkable result of the Rumelhart et al (1985) procedure is that it yields an elegant solution to this problem.

The learning task of the network is to minimise the error in the output functions for all the input patterns presented to it. The error function that must be minimised is shown below where \( E \) denotes error, \( p \) is a subscript denoting pattern, \( t \) denotes the target value, and \( y \) stands for the output activation.

\[ E = \frac{1}{2} \sum_p \sum_j (t_pv_j - y_{pj})^2 \]  

B.2

To minimise \( E \) by gradient descent, the partial derivative of \( E \) with respect to each weight in the network must be computed. Taking a particular pattern \( p \), the partial derivative is then computed in two passes, a forward pass, and a backward pass which propagates the derivatives back through the layers, hence the reason why the approach is called ‘backpropagation of error’. The backward pass begins by differentiating B.2 above, yielding:

\[ \frac{\partial E}{\partial y_{pj}} = t_{pj} y_{pj} \]  

B.3
The chain rule is then invoked to compute

$$\frac{\partial E_p}{\partial \text{Net}_{pj}} = \frac{\partial E_p}{\partial y_p} \cdot \frac{\partial y_p}{\partial \text{Net}_{pj}}$$  \hspace{1cm} \text{B.4}$$

The second term of the above equation is produced by differentiating B.1 which yields

$$\frac{\partial a_p}{\partial \text{Net}_{pj}} = f'(\text{Net}_{pj}) = y_p(1-y_p)$$  \hspace{1cm} \text{B.5}$$

Hence the effect of the error due to a change in the total input to an output unit is known. But, as the total output is simply a linear function of the output from previous layers and the related connection weights, the effect on the error due to a change in the previous outputs and weights can be obtained. For a weight $w_{ij}$ from unit $i$ to unit $j$, the derivative is

$$\frac{\partial E_p}{\partial w_{ij}} = \frac{\partial E_p}{\partial \text{Net}_{pj}} \cdot y_p$$  \hspace{1cm} \text{B.6}$$

and the effect of all the connections emanating from unit $i$ is simply

$$\frac{\partial E_p}{\partial y_p} = \sum \frac{\partial E_p}{\partial \text{Net}_{pj}} \cdot w_{ij}$$  \hspace{1cm} \text{B.7}$$

We can now define two different error signals depending on whether the unit is an output or a hidden unit. For an output unit the error signal used to change the network is

$$\delta_{pj} = (t_p - y_p) f'(\text{Net}_{pj})$$  \hspace{1cm} \text{B.8}$$

and for a hidden unit the network is modified by

$$\delta_{pj} = f'(\text{Net}_{pj}) \sum \delta_{pj} w_{ij}$$  \hspace{1cm} \text{B.9}$$

Hence the weights in the networks are changed by

$$\Delta_p w_{ij} = \eta \delta_{pj} y_p$$  \hspace{1cm} \text{B.10}$$

where $\eta$ is a learning parameter which is used to scale the weight change.
APPENDIX C

Lakoff’s argument against objectivist metaphysics

Lakoff presents two arguments against the classical account of semantics, arguing that it is based on an objectivist metaphysics. The two arguments are reconstructed below.

The empirical argument

This argument is developed in Lakoff (1987, Chapter 3). He mentions inter alia the following results:

- The conjunction of blue and green is perceived to be turquoise, but no object can be seen as simultaneously red and green; this colour mixture is perceived as murky brown. This is because red and green result from opposing responses of the same neurons, whereas blue and green derive from responses of different neurons. In the case of colour perception then, the architecture of the nervous system plays a role in the categorisation of reality.

- In terms of the objectivist account it can be expected that categories that are the easiest to process, and physiologically the most simple, will constitute conceptual primitives. Basic level categories are intermediate in terms of taxonomic hierarchies, and have a great deal of internal structure (Rosch, 1973). Yet humans find them easiest to process (e.g. Rosch, 1977). Categories which would seem to be cognitively complex from a purely objectivist point of view thus turn out to be cognitively simple.

- The organisation of semantic categories into different hierarchical levels in the conceptual system cannot easily be accounted for in terms of the objectivist paradigm. Aspects such as part-whole organisation, imaging capacity, motor organisation, and knowledge organisation play a dominant role here and these are not given by ‘objective’ reality, but are mediated by cognitive/physiological factors.

The logical argument

This argument is developed in Lakoff (1987, Chapter 5). It is borrowed from Putnam (1980) who first remarked that results obtained from the Lowenheim Skolem theorem have unexpected implications for the
use of model theory in semantics. Lakoff observes that the development of the objectivist paradigm in semantics is intimately tied to the use of mathematical logic, as exemplified by the use of model-theoretic semantics (henceforth MTS) to characterise the meaning of natural language. However, following Putnam, Lakoff purports to show that MTS is internally inconsistent. The inconsistency derives from the following two claims which are not both true:

- Semantics characterises the way symbols relate to the world; and
- Semantics characterises meaning.

The inconsistency is demonstrated by focussing on two basic assumptions adopted in MTS:

- The meaning of a sentence is a function which assigns a truth value to that sentence in each possible world;
- The meaning of parts of a sentence cannot be changed significantly without affecting the meaning of the whole sentence.

Putnam's crucial point is that MTS has no means of securing the reference of its symbols. Different models are compatible with the same referents so that reference is not fixed in any unique way. The implication is consequently that the reference of the parts of a sentence can be changed while preserving the truth for the whole sentence. Recall, that Fodor and Pylyshyn (1988) stress that it is precisely the fact that the classical symbolic account explains how meanings can be composed recursively in natural language, that makes it such a powerful theoretical framework. Hence, if it can be shown that the semantic account underlying the classical approach is problematic, the force of their argument against ANN models of language is weakened. To show that the classical, objectivist account of meaning drawing from MTS is problematic, Lakoff proceeds as follows. In MTS semantic interpretation is conceptualised as truth within a model. Thus, because it can be shown that truth value (i.e. satisfaction within a model) can remain constant while the extensional (i.e. the set of things denoted by a word) and intensional (i.e. properties associated with the word) definitions of the words in a sentence are changed, assumption (2) above is violated.

Putnam has given a formal, technical proof and an informal illustration of the method of proof. Lakoff (1987) is concerned with the presentation of the latter. It is summarised below.

Consider the sentence:
(1) A cat is on a mat.

On the standard interpretation the sentence is true in all possible worlds where there is at least one cat on at least one mat in some time in the past, present, or future. It is furthermore accepted that cat refers to the creature ‘cat’ and that mat refers to the floor covering ‘mat’. Putnam’s aim is to show that both terms can be drastically reinterpreted so that in the actual world cat refers to ‘cherries’ and mat to ‘trees’ without changing the truth value of the sentence in any possible world. To this end, he formulates a second sentence:

(2) A cat* is on a mat*.

The properties of cat* and mat* are given by three cases:

(a) Some cat is on some mat, and some cherry is on some tree.
(b) Some cat is on some mat, and no cherry is on any tree.
(c) Neither of the above.

The definitions of the new terms are:

**Definition of cat***

x* is a cat iff (i.e. if and only if) case (a) holds and x is a cherry; or case (b) holds and x is a cat; or case (c) holds and x is a cherry.

**Definition of mat***

x is a mat* iff case (a) holds and x is a tree; or case (b) holds and x is a mat; or case (c) holds and x is a quark.

It can now be shown that providing these new intensions (i.e. definitions of the properties associated with the words) does not alter the MTS interpretation of the sentence. In other words, (1) and (2) have exactly the same truth value in any possible world. This is because in case (a) both sentences have the same truth conditions; all cherries are cats* and all trees are mats* in worlds of this kind. In the actual world, some cherry is on some tree, and the actual world is consequently a world of this kind. In this world, cat* refers to cherries and mat* refers to trees. In worlds satisfying case (b) both sentences have the same truth values because cat/cat* and mat/mat* are coextensive terms. The point here is that although cats are cats* in some worlds - those satisfying (b) - they are not cats in the actual world. In worlds falling under (c), sentence (1)
is false and sentence (2) is likewise false (because a cherry can’t be on a quark).

The proof thus shows that reinterpreting the terms of (1) by assigning radically new intensions to the words in the sentence, preserves truth values in every model.