

**Connectionist Neuropsychology:
The Breakdown and Recovery of Behavior
in Lesioned Attractor Networks**

David C. Plaut

September 1991

CMU-CS-91-185

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Thesis Committee:

Geoffrey Hinton, chair

David Touretzky

Tom Mitchell

James McClelland

Martha Farah, CMU Psychology

Copyright © 1991 David C. Plaut

This research was supported by grant 87-2-36 from the Alfred P. Sloan Foundation. All of the simulations described in this thesis were run on a Silicon Graphics Iris-4D/240S using the Xerion simulator developed by Tony Plate.

Abstract

An often-cited advantage of connectionist networks is that they degrade gracefully under damage. Most demonstrations of the effects of damage and subsequent relearning in these networks have only looked at very general measures of performance. More recent studies suggest that damage in connectionist networks can reproduce the specific patterns of behavior of patients with neurological damage, supporting the claim that these networks provide insight into the neural implementation of cognitive processes. However, the existing demonstrations are not very general, and there is little understanding of what underlying principles are responsible for the results. This thesis investigates the effects of damage in connectionist networks in order to analyze their behavior more thoroughly and assess their effectiveness and generality in reproducing neuropsychological phenomena.

We focus on connectionist networks that make familiar patterns of activity into stable “attractors.” Unit interactions cause similar but unfamiliar patterns to move towards the nearest familiar pattern, providing a type of “clean-up.” In unstructured tasks, in which inputs and outputs are arbitrarily related, the boundaries between attractors can help “pull apart” very similar inputs into very different final patterns. Errors arise when damage causes the network to settle into a neighboring but incorrect attractor. In this way, the pattern of errors produced by the damaged network reflects the layout of the attractors that develop through learning.

In a series of simulations in the domain of reading via meaning, networks are trained to pronounce written words via a simplified representation of their semantics. This task is unstructured in the sense that there is no intrinsic relationship between a word and its meaning. Under damage, the networks produce errors that show a distribution of visual and semantic influences quite similar to that of brain-injured patients with “deep dyslexia.” Further simulations replicate other characteristics of these patients, including additional error types, better performance on concrete vs. abstract words, preserved lexical decision, and greater confidence in visual vs. semantic errors. A range of network architectures and learning procedures produce qualitatively similar results, demonstrating that the layout of attractors depends more on the nature of the task than on the architectural details of the network that enable the attractors to develop.

Additional simulations address issues in relearning after damage: the speed of recovery, degree of generalization, and strategies for optimizing recovery. Relative differences in the degree of relearning and generalization for different network lesion locations can be understood in terms of the amount of structure in the subtasks performed by parts of the network.

Finally, in the related domain of object recognition, a similar network is trained to generate semantic representations of objects from high-level visual representations. In addition to the standard weights, the network has correlational weights useful for implementing short-term associative memory. Under damage, the network exhibits the complex semantic and perseverative effects of patients with a visual naming disorder known as “optic aphasia,” in which previously presented

objects influence the response to the current object. Like optic aphasics, the network produces predominantly semantic rather than visual errors because, in contrast to reading, there is some structure in the mapping from visual to semantic representations for objects.

Taken together, the results of the thesis demonstrate that the breakdown and recovery of behavior in lesioned attractor networks reproduces specific neuropsychological phenomena by virtue of the way the structure of a task shapes the layout of attractors.

To Angela, Bern, and Benli.

Acknowledgements

First of all, I'd like to thank my advisor, Geoff Hinton, for sharing with me his enthusiasm for research and seemingly boundless wealth of ideas and insights. It is truly an amazing experience working with him, and I hope I have absorbed even a fraction of what he has offered me. I also want to thank him for his support, guidance, and quiet confidence in me during what might otherwise have been a difficult arrangement when he moved to the University of Toronto after my second year at CMU.

I'd also like to thank the other members of my thesis committee. Dave Touretzky looked after me in Geoff's absence, keeping me on track as I navigated through the vagaries of research from proposal to thesis. Martha Farah encouraged my initial explorations into cognitive neuropsychology, and showed me the value of clear explanation in connectionist modeling. Tom Mitchell brought a fresh perspective to the work and helped me embed it in a broader context. Finally, Jay McClelland welcomed me into his research group and into the excitement and challenge of cognitive modeling. Perhaps most importantly, he simply insisted that I finish. His encouragement and effort in fostering the next stage of my career are deeply appreciated.

Although not formally a member of my committee, Tim Shallice has certainly had the most profound impact on the thesis work itself, and perhaps on my direction as a researcher as well. Tim's breadth of knowledge and intuitive wisdom are remarkable, and our work together has been thoroughly rewarding for me. The experience of such a rich interdisciplinary collaboration has crystallized for me the style of work I hope to pursue throughout my career, and I look forward to continuing our work together.

In addition to Martha, Jay, and Tim, three other researchers have contributed substantially in helping me bridge the somewhat murky waters between computer science and psychology. I am deeply indebted to David Taylor for first encouraging me to pursue work in cognitive science as an undergraduate, and for later providing me with a remarkable opportunity to engage in both psychological research and computer simulation work. Paddy McMullen bandied issues in object recognition with me over many a lunch and dinner, continually bringing me back to the psychological details. Marlene Behrmann wrestled with me about the role of connectionist modeling in psychological theorizing, her skepticism wonderfully complemented by her interest and (dare I say it) enthusiasm. Yet far more valuable to me than the intellectual contributions are the deep and lasting friendships that have grown from these interactions.

Although David Taylor introduced me to connectionist research, I thank Jerry Feldman for taking my initial forays seriously enough to guide me in focusing my interests more productively. I thank the many members of the CMU Boltzmann research group who have contributed to my work in important ways. Mark Derthick was a valued comrade in the struggle between connectionist and symbolic AI, helping me see the substantial commonalities. Rick Szeliski joined me in trying

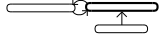
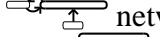
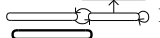
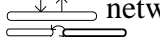
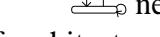
to bring together computational and biological perspectives in vision research. In addition, I'd like to specifically thank David Ackley, Scott Fahlman, Roni Rosenfeld, and Dave Touretzky for contributing to an interesting and productive research environment. I also thank the members of the CMU PDP research group and the Toronto Connectionist research group, particularly Rich Zemel and Sue Becker, for welcoming me amongst their ranks. Tony Plate deserves special thanks for his help with the Xerion simulator.

I could not have completed a Ph.D. without the support and encouragement of my family and friends. My parents have been completely supportive of all of my academic and personal endeavors, and whatever I have achieved has come out of the context they provide. In addition to Paddy, Marlene and Mark, friends such as Lynn Baumeister, Chris Tague, and Violetta Cavalli-Sforza have made my life richer and more rewarding.

Finally, I must express my deepest gratitude to Angela Hickman, Bernadette Kowalski Minton, and Benjamin Pierce. We banded together as entering graduate students six years ago and have seen each other through everything ever since. I cannot imagine greater friends. Angela, Bern, and Benli have shared it all with me and have made it all possible, and it is to them that I dedicate this thesis.

Contents

1	Introduction	1
1.1	Connectionist networks	1
1.2	Attractors	2
1.3	Damage in connectionist networks	4
1.4	Connectionist neuropsychology	6
1.5	Thesis overview	7
2	Connectionist modeling in neuropsychology	11
2.1	Cognitive neuropsychology	11
2.1.1	Modularity and dissociations	12
2.1.2	Acquired dyslexia	13
2.2	Computational modeling in cognitive neuropsychology	17
2.2.1	Conventional implementations	17
2.2.2	Connectionist approaches	20
2.3	Connectionist modeling of acquired dyslexia	21
2.3.1	Neglect and attentional dyslexias	22
2.3.2	Surface dyslexia	25
2.3.3	Deep dyslexia	28
2.4	Deep dyslexia	29
2.5	Motivation of a connectionist account	32
2.6	A preliminary connectionist model of deep dyslexia	33
2.6.1	The task	33
2.6.2	The network	37
2.6.3	The training procedure	38
2.6.4	The testing procedure	38
2.6.5	Attractors	41
2.7	Evaluation of the model	43
2.7.1	The task	44
2.7.2	The network	45
2.7.3	The training procedure	45
2.7.4	The testing procedure	45
3	Response generation: Mapping semantics to phonology	47
3.1	Phonological blends	48
3.1.1	The task	48

3.1.2	The network	48
3.1.3	The training procedure	49
3.1.4	The effects of lesions	50
3.1.5	An explanation for blends	55
3.2	Eliminating blends	56
3.2.1	The network architecture	57
3.2.2	The training procedure	57
3.2.3	The effects of lesions	60
3.3	Comparison with response criteria	63
3.4	Impairments in mapping semantics to phonology	65
3.5	Summary	68
4	The relevance of network architecture	69
4.1	Alternative architectures	72
4.2	The task	75
4.3	The training procedure	75
4.4	The effects of lesions	76
4.4.1	The  network	77
4.4.2	The  network	81
4.4.3	The  network	84
4.4.4	The  network	89
4.4.5	The  network	95
4.5	Summary of architecture comparisons	97
4.5.1	Generality of the H&S findings	97
4.5.2	The strength of attractors	100
4.5.3	Error types	101
4.6	Item- and category-specific effects	102
4.7	Definitions of visual and semantic similarity	109
4.8	Visual-then-semantic errors	111
4.9	Effects of lesion severity on error type	116
4.10	Error patterns for individual lesions	117
4.11	Summary	119
5	The relevance of learning procedure	121
5.1	Deterministic Boltzmann Machines	123
5.1.1	Energy minimization	124
5.1.2	Simulated annealing	125
5.1.3	Contrastive Hebbian learning	126
5.1.4	The task	128
5.1.5	The network architecture	129
5.1.6	The training procedure	129
5.1.7	The effects of lesions	134
5.2	GRAIN networks	139
5.2.1	The training procedure	142
5.2.2	The effects of lesions	142

5.3	Confidence in visual vs. semantic errors	145
5.4	Lexical decision	147
5.5	Summary	149
6	Extending the task domain: Effects of abstractness	151
6.1	Effects of abstractness in deep dyslexia	152
6.2	A semantic representation for concrete and abstract words	153
6.3	Mapping orthography to semantics	154
6.4	Mapping semantics to phonology	158
6.5	The effects of lesions	161
6.6	Network analysis	169
6.7	Summary	177
7	Relearning after damage	179
7.1	Cognitive remediation of acquired dyslexia	180
7.2	Previous studies of relearning in networks	182
7.3	Experiments in relearning after damage	186
7.3.1	The training procedure	186
7.3.2	The effects of lesions	187
7.3.3	The relearning procedure	187
7.3.4	Pre-attractor lesions	189
7.3.5	Within-attractor lesions	202
7.3.6	An explanation for differences in relearning and generalization	214
7.3.7	Comparisons with patient rehabilitation studies	217
7.4	Designing therapy to maximize generalization	217
7.4.1	Semantic prototypicality	218
7.4.2	Selecting retraining items based on prototypicality	219
7.4.3	Effects of prototypicality on generalization	220
7.4.4	An explanation for the prototypicality effects	220
7.4.5	A more detailed test of prototypicality effects	224
7.5	Summary	226
8	Visual object naming in optic aphasia	227
8.1	Optic aphasia	228
8.1.1	Visual agnosia	229
8.1.2	Optic aphasia	229
8.1.3	Theoretical accounts	232
8.2	Short-term correlational weights	235
8.3	A simulation of visual object naming in optic aphasia	237
8.3.1	The task	237
8.3.2	The network	248
8.3.3	The training procedure	249
8.3.4	The lesioning procedure	249
8.3.5	Correct performance	250
8.3.6	Horizontal errors	250

8.3.7	Vertical errors	254
8.3.8	Effects of type of response to the preceding object	257
8.3.9	Effects of lesion location	259
8.3.10	Item effects	261
8.4	Relation to deep dyslexia simulations	268
8.5	Recognition in optic aphasia	270
8.6	Summary	272
9	General discussion	273
9.1	Computational generality	274
9.1.1	Response generation	274
9.1.2	The importance of attractors	275
9.2	Empirical adequacy	278
9.2.1	Extensions of the Hinton & Shallice results	278
9.2.2	Remaining empirical issues	282
9.3	Theoretical issues	287
9.3.1	The right hemisphere theory	288
9.3.2	Attractors vs. logogens	289
9.4	Extensions of the approach	291
9.4.1	Other “deep” syndromes	291
9.4.2	Cognitive remediation	292
9.4.3	Optic aphasia	293
9.5	The impact of connectionist modeling in neuropsychology	294
9.6	The impact of neuropsychology on connectionist modeling	296
9.7	Future work	299
9.7.1	Implementing a dual-route model of reading	299
9.7.2	Rehabilitation study	300
9.7.3	Modality-specific semantic impairments	300
9.7.4	Pure alexia	301
9.8	Conclusion	301
10	Back-propagation through time	303
A	The units	303
B	The forward pass	303
C	The error function	303
D	The backward pass	304
E	Weight updating	304
F	Training criterion	304

Chapter 1

Introduction

1.1 Connectionist networks

Connectionist networks, also known as neural networks or parallel distributed processing (PDP) networks, are becoming increasingly influential in artificial intelligence and psychology as a methodology for developing computational models of cognitive processes. In contrast to conventional programs that compute via the sequential application of stored commands, these networks compute via the parallel cooperative and competitive interactions of a large number of simple neuron-like processing units. Each unit has an associated activity level, or “state,” typically ranging between 0 and 1. Positive or negative real-valued “weights” on connections between units modulate how the units interact. In most formulations, the total input to a unit is simply the weighted sum of the states of units from which it receives connections; its own state is a smooth, non-linear function of this total input. All of the long-term knowledge of the system is encoded in the weights; learning involves modifying the weights to improve performance on some task.

In performing a task, input is presented to the network by setting the states of some designated “input” units. The remaining units then update their states to be maximally consistent with each other, and with the input, given the knowledge encoded in the weights. The resulting pattern of activity across all of the units constitutes the network’s interpretation of the input. The states of designated “output” units represent the response of the network to the input. The input and output units are called “visible” because their correct states are determined by the environment external to the network; any remaining units are thus “hidden.” Because the environment does not specify the states of hidden units, the learning procedure must develop internal representations over these units that are useful for solving the task.

A number of useful properties arise naturally out of this style of computation.

- The ability to bring a massive amount of knowledge to bear simultaneously in determining the best interpretation of any input (Ballard et al., 1983).
- Reconstructive content-addressable memory (Hinton & Anderson, 1981; Pollack, 1990).

- Automatic similarity-based generalization and a smooth transition from exceptions to regularities (Hinton et al., 1986; McClelland & Rumelhart, 1985).
- Natural integration of multiple sources of information, such as top-down and bottom-up (McClelland & Rumelhart, 1981).
- The ability to continually improve performance and build useful internal representations using a local learning rule (Hinton, 1989a).
- Robustness in the presence of noisy, incomplete, or partially inconsistent information (Derthick, 1988) and graceful degradation with damage (Hinton & Sejnowski, 1986; Smolensky, 1986).
- Rapid relearning after damage and spontaneous recovery of unrehearsed knowledge (Hinton & Plaut, 1987; Hinton & Sejnowski, 1986).
- A straightforward and efficient implementation in VLSI (Alspector & Allen, 1987; Mead, 1989) or massively parallel hardware (Pomerleau et al., 1988; Zhang et al., 1990).

Connectionist networks have been successfully applied to problems in a wide range of cognitive domains, such as high-level vision and attention, learning and memory, reading and language, speech recognition and production, and sequential reasoning (see McClelland et al., 1986, and recent Cognitive Science Society conference proceedings).

1.2 Attractors

In some connectionist networks, the units are organized into a sequence of layers such that units in later layers receive connections only from units in earlier layers. This type of “feed-forward” architecture has the advantage that each unit need only compute its state once in processing an input, but has the disadvantage that the possible ways in which units can interact are severely restricted. In contrast, more complex, “recurrent” networks have no restrictions on how units can be connected, enabling interactions among units within a layer, and feedback from later to earlier layers. When presented with input, the units must update their states repeatedly, because changing the state of a unit may change the *input* to earlier units. Each unit computes a new state at every time step, called an “iteration.” As a result, the pattern of activity over the entire network changes over time in response to a fixed input. The network must learn to gradually settle down into the appropriate final pattern of activity that corresponds to the correct interpretation of the input.

This process can be conceptualized as movement in a multi-dimensional space that has a dimension for the state of each unit in the network. At any instant, the current pattern of activity of the network is represented as a particular point in this “state” space. When an input is presented to the network, the initial pattern of states of all of the units constitutes the starting point. As units update their states, the point representing the current pattern of activity moves in state space,

eventually arriving at the point corresponding to the correct interpretation of the input. In fact, there is a region in state space around this final point such that, if the network starts anywhere within this region, it will still settle into the same final pattern of activity. The point corresponding to this pattern is called an “attractor” in state space, and the surrounding region is called its “basin” of attraction. The shapes and positions of the basins depend on the ways that units interact, which in turn depend on the connection weights. Thus learning in a recurrent network amounts to building and shaping the attractor basins so that the network settles to the appropriate attractor when started at the initial pattern of activity corresponding to each input.

In addition to “point” attractors, recurrent networks can be trained to settle into “limit cycle” attractors, in which the network’s activity repeatedly moves through a fixed trajectory in state space (Pearlmutter, 1989). Recurrent networks can also develop “chaotic” attractors, in which the trajectory does not repeat but is constrained in its complexity (Skarda & Freeman, 1987). However, in this thesis we will only be concerned with networks that settle to point attractors.

The usefulness of attractors becomes apparent when we consider how a recurrent network with a given set of weights maps initial activity patterns onto final activity patterns. All of the patterns within a particular basin of attraction map to the same attractor pattern. Since points that are nearby each other in state space represent similar patterns of activity, the operation of the attractor can be thought of as a kind of similarity-based categorization. Thus, the fundamental property of attractors is that they carve up the large, typically continuous space of all possible activity patterns into a much smaller discrete set of (attractor) patterns on the basis of similarity. For this reason, attractors are *not* appropriate for tasks, such as continuous function approximation (Lapedes & Farber, 1987), in which slight differences in input must be maintained to produce slight differences in output. However, the ability of attractors to give the same output to similar inputs is quite useful in many situations.

For example, consider a network that has learned to make some number of patterns into attractors. Suppose we present the network with a noisy or incomplete version of one of these patterns. As long as the corrupted pattern is more similar to the original than to any other (i.e. falls within the appropriate attractor basin), the operation of the network in settling will “clean-up” or “complete” the pattern (see Figure 1.1). If each pattern is composed of separate types of information to be associated, such as a name with a face, then the ability of the network to reconstruct one part given another can be used to implement associative or “content-addressable” memory (Hinton & Anderson, 1981). The network can also be thought of as generalizing to novel inputs that fall within the basin of attraction of a familiar input. Whether this generalization is appropriate will depend on the definition of the task and on details of the layout of attractor basins.

In addition to giving the same interpretation to similar inputs, attractors are also useful when similar inputs require very *different* interpretations (points **A** and **B** in Figure 1.1). An example of this is in understanding written words, in which it is common that very visually similar words

Figure 1.1: A depiction of attractors in state space (left), and a demonstration of their ability to clean-up or complete corrupted patterns (right) (from Hertz et al., 1991, pp. 12–13).

(e.g. CAR and CAP) must produce completely different meanings. This is difficult for connectionist networks in which the representation of an entity is distributed over a number of units (Hinton et al., 1986). The reason is that other units are influenced by the input on the basis of a simple weighted sum of unit activities. Since similar inputs are represented by similar patterns of activity, they produce similar summed input to other units. In a feed-forward network, the bias to give similar interpretations to similar inputs can be overcome either by introducing many layers of non-linear units between the input and output, or by having very large weights between units. Both of these approaches require very long learning times. In a recurrent network, unit non-linearities are reapplied at every time step. Thus initially small state differences can be magnified into quite large ones as the network settles. In essence, the network can learn to position the boundaries between attractor basins to “pull apart” similar initial patterns of activity so they settle to possibly quite distant final patterns.

1.3 Damage in connectionist networks

The weight changes that occur during learning modify how units interact, determining which patterns are attractors and which other patterns settle into them. In this way, learning has the effect of positioning and shaping the attractor basins in state space. Hence, we speak of a network as developing or “building” attractors over the course of learning.

If a recurrent network has learned a task, we know that it has developed attractors for each of the output patterns in the task, and that the initial activity corresponding to each input pattern falls somewhere within the basin of attraction of the corresponding output pattern. However, we know little else about the layout of attractor basins in state space—what the shapes of the basins for trained patterns are like, and whether there are additional, “spurious” attractors. Also, we have little understanding of what aspects of the design of the network most strongly influence the nature of the attractors it develops.

One might be tempted to map out the entire state space, by starting the network in the pattern of activity corresponding to each sampled point in the space, and then noting what pattern the network settles into. Unfortunately, not only would such a procedure be computationally intractable, but the results would be in a form that contributed little to our understanding of the principles that influence the organization of the attractors. A more restricted approach would be to systematically test the generalization of the network to inputs that are similar to the ones on which it has been trained. However, this approach is limited in that it only provides information about regions of state space near the attractors for familiar patterns, but not about other attractors that may have developed during learning.

Another way to understand the layout of attractors in a network is to study its behavior under damage. Damaging a connectionist network typically involves removing some of the units and/or connections, or adding random noise to the weights. Attractors make a network more robust to damage in the same way as they clean-up noisy or incomplete input. In fact, corrupted input can be interpreted as damage to the input layer of the network. This damage changes the initial pattern of activity, moving the starting point of the network in state space. As long as this point still falls within the appropriate basin of attraction, the network behaves normally. However, if the noise or damage moves the starting point outside of the basin of the correct attractor, the operation of network will “clean-up” the initial pattern into the final pattern corresponding to some other attractor. This can be seen in Figure 1.1 by imagining that the dotted line between **A** and **B** (representing a boundary between two attractor basins) moves to the other side of **A**. The new final pattern for **A** may correspond to another familiar attractor, or it may correspond to an unfamiliar “spurious” attractor. In either case, the network has misinterpreted the input and will generate an incorrect response. If the damage is random, the likelihood of particular incorrect responses to some input depends on the shapes and positions of the attractor basins in state space. In this way, the pattern of errors produced by the damaged network reflects the layout of the attractors that develop through learning.

In actuality, the effects of damage to units and connections internal to the network are more complex than for damage to input units. If the damaged units and connections are involved in the interactions that implement the attractors, then damaging them corrupts the layout of the attractors themselves (the dotted lines in Figure 1.1). Some attractors may disappear, others may be created,

and the boundaries between existing attractors may move. These modifications can also cause an input to fall within the basin of attraction of an inappropriate attractor. Since the shape of an attractor influences the degree to which movement of its boundaries can “capture” other inputs, the pattern of errors produced by the network under this type of damage also reflects the nature of the attractors in state space. However, in this case the relationship of the resulting error pattern to the original layout of attractor basins is less straightforward than when only the input is corrupted.

1.4 Connectionist neuropsychology

Beyond their computational properties, a major attraction of using connectionist networks to model cognitive processes is that their general similarity to neurobiology suggests that the nature of computation in these networks may provide insight into how cognitive processes are implemented in the brain (Sejnowski et al., 1989). Evidence that is often put forward in support of this claim is that, like brains, connectionist networks degrade gracefully under damage. That is, with partial damage, the network’s performance on a task is only partially impaired rather than being completely lost. However, most demonstrations of this property have only considered the effects of damage on very general measures of performance, such as total error on a task. The relevance of connectionist modeling to the neural implementation of cognitive processes would be far better established if it were shown that the detailed pattern of breakdown and recovery of behavior in damaged connectionist networks resembles that of patients with cognitive impairments due to neurological damage.

Some particularly encouraging preliminary work in this direction involves modeling the cognitive deficits of certain types of patients with brain damage by “lesioning” a connectionist model of the normal process (Farah & McClelland, 1991; Hinton & Shallice, 1991; Mozer & Behrmann, 1990; Patterson et al., 1990). Most of these initial investigations have focused on deficits in word reading, known as “acquired dyslexias.” The work of Mozer & Behrmann (1990) and Patterson et al. (1990) followed the standard approach in cognitive neuropsychology of using models of the normal reading process to account for disorders of reading as a result of damage. The design of each model was motivated to account for normal performance and not fundamentally altered in accounting for patient data. The lesion simulations provide independent validation of the properties of the models that enable them to reproduce phenomena they were not initially designed to address. In this way, these approaches avoid the criticism, often leveled against connectionist models (e.g. Massaro, 1988), that their success at reproducing psychological data tells us little if anything about human cognition because the models are so underconstrained.

The work of Hinton & Shallice (1991) is somewhat different in nature. They were primarily concerned with “deep dyslexia,” an acquired reading disorder in which patients can only pronounce written words via their meaning, and occasionally make errors in this process (e.g. misreading the

word RIVER as “ocean”). These patients also show visual influences in their errors (e.g. SWORD \Rightarrow “words”), and a range of other symptoms. In modeling deep dyslexia, Hinton & Shallice were less concerned with supporting a particular model of normal word comprehension performance, in part because it is less feasible given our current limited understanding of lexical semantics. Rather, their goal was more general: to investigate the effects of damage on the behavior of a more general type of network that maps from strings of letters to semantics. By not being tied to a particular normal model, it becomes possible to systematically explore the space of models that qualitatively reproduce the neuropsychological phenomena, allowing the relevance and implications of various design decisions to be evaluated. To the extent that these design decisions arise out of general connectionist principles, an understanding of their impact in the particular domain of acquired dyslexia sheds light on the overall enterprise of connectionist neuropsychology. To this end, a major focus of this thesis is an empirical investigation of the major design decisions of the Hinton & Shallice model, aimed at clarifying and improving the properties of the model that led to its successes, limitations and failures.

1.5 Thesis overview

This thesis investigates the breakdown and recovery of behavior in lesioned attractor networks, in order to analyze their behavior more thoroughly and identify the computational principles that enable them to reproduce detailed neuropsychological phenomena. The term “connectionist neuropsychology” is intended to apply to the thesis research in two ways. The first is in doing cognitive neuropsychology *with* connectionist networks—using them to better understand patient behavior. The second is in doing cognitive neuropsychology *on* connectionist networks—analyzing their impaired behavior under damage as a way of gaining insight into the nature of their own representations and processes. An important insight that emerges from the thesis is that the structure of a task has a profound influence on the nature of the layout of attractors in state space. The organization of the thesis is as follows:

Chapter 2. A general overview of the cognitive neuropsychology of reading is presented, emphasizing the reading behavior of deep dyslexics, who are the focus of much of the simulation work in the thesis. After a brief review of related efforts in modeling neuropsychological phenomena, a summary and critical analysis of the Hinton & Shallice model is presented. Each of the four major types of design decisions that went into developing the model, relating to the task, the network architecture, the training procedure, and the testing procedure, motivates the research in one of the four subsequent chapters.

Chapter 3. The Hinton & Shallice model was quite limited in the way it generated responses, resorting to external criteria applied directly to semantic representations. A straightforward extension of the model to generate explicit pronunciations from semantics results in the inappropriate

production of novel “blends” of familiar responses under damage. Simulations in this chapter illustrate these difficulties and demonstrate techniques for overcoming them.

Chapter 4. Aspects of the design of the network architecture that Hinton & Shallice used were rather inelegant. In this chapter we investigate the effects of damage in a range of alternative network architectures, each intended to evaluate the relevance of a particular aspect of the original network. The results demonstrate that the existence of attractors is critical for reproducing the error pattern of deep dyslexics, but that the architectural details that enable them to develop are relatively unimportant. Additional simulations serve to verify that the general effects produced by the networks aren’t due to idiosyncratic characteristics of the word set or interpretation procedure. They also demonstrate that the networks behave like deep dyslexic patients in terms of the pattern of responses after individual lesions.

Chapter 5. The Hinton & Shallice model was trained with a powerful but biologically implausible learning procedure known as “back-propagation through time.” Simulations in this chapter replicate the behavior of deep dyslexics using the more plausible learning procedure of contrastive Hebbian learning in a deterministic Boltzmann Machine (DBM). A closely-related stochastic GRAIN network is also developed and compared with the deterministic one. In addition to being more plausible as procedures that might underly human learning, both DBM and GRAIN networks have interesting computational characteristics not shared by back-propagation networks. We conclude by demonstrating how these characteristics are useful for understanding two aspects of deep dyslexic reading behavior: differences in confidence in error responses, and the preserved ability to distinguish words from non-words.

Chapter 6. The final design issue that directly concerns the Hinton & Shallice model is the definition of the task of reading via meaning. The original training set was too limited to address important issues known to influence the reading behavior of deep dyslexics. Specifically, in patients abstract words are read more poorly than concrete words, and are particularly prone to visual errors. We reproduce these effects in a network that can pronounce both concrete and abstract words via their semantics, defined so that abstract words have fewer semantic features. Surprisingly, severe damage within semantics produces the *opposite* effects, with concrete words read more poorly, similar to a particular, enigmatic patient with “concrete word dyslexia.”

Chapter 7. A major motivation for many cognitive neuropsychologists is that a more detailed analysis of the breakdown of cognitive mechanisms due to brain damage may lead to the design of more effective therapy to remediate these impairments. We attempt to extend the relevance and usefulness of connectionist modeling in neuropsychology to address issues in the rehabilitation of cognitive deficits following brain damage: the speed of recovery, degree of generalization, and strategies for optimizing recovery. Two of the dyslexic networks are retrained on a subset of the words after damage, and then tested for their performance on the remaining words. Relative differences in the degree of relearning and generalization for different lesion locations can be

understood in terms of the amount of structure in the subtasks performed by parts of the network. Furthermore, retraining on words whose semantics are atypical of their category yields more generalization to more prototypical words than *vice versa*, although the word set is too limited to support definitive implications for patient therapy.

Chapter 8. This chapter demonstrates the generality of the principles that help explain the reading deficits in deep dyslexia, by extending them to account for the characteristics of another syndrome in the related domain of visual object recognition and naming. A similar network is trained to generate semantic representations of objects from high-level visual representations. In addition to the standard weights, the network has more rapidly changing correlational weights useful for implementing short-term associative memory. Under damage, the network exhibits the complex semantic and perseverative effects of patients with a visual naming disorder known as “optic aphasia,” in which previously presented objects meaningfully influence the response to the current object. The greater structure in mapping visual to semantic representations for objects vs. words explains why the errors of optic aphasics are predominantly semantic rather than visual.

Chapter 9. The final chapter summarizes the thesis, evaluating the computational generality and empirical adequacy of using attractor networks to model deep dyslexia, relearning after damage, and optic aphasia. The general impact of connectionist modeling in neuropsychology is discussed, as well as how investigations of the effects of damage provide unique insights into the representational and computational properties of connectionist networks. It is concluded that the breakdown and recovery of behavior in lesioned attractor networks reproduces detailed neuropsychological phenomena by virtue of the way the structure of a task shapes the layout of attractors.

