SELF-ORGANIZING NETWORKS FOR GROUNDED LANGUAGE ACQUISITION

A SELF-ORGANIZING COMPUTATIONAL NEURAL NETWORK ARCHITECTURE WITH APPLICATIONS TO SENSORIMOTOR GROUNDED LINGUISTIC GRAMMAR ACQUISITION

By

PETER JANSEN, B.I.S.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (2010) (Psychology)	McMaster University Hamilton, Ontario

TITLE: A Self-organizing Computational Neural Network Architecture with Applications to Sensorimotor Grounded Linguistic Grammar Acquisition

AUTHOR: Peter Jansen, B.I.S. (University of Waterloo)

SUPERVISOR: Professor Scott Watter

NUMBER OF PAGES: ix, 169

Abstract

Connectionist models of language acquisition typically have difficulty with systematicity, or the ability for the network to generalize its limited experience with language to novel utterances. In this way, connectionist systems learning grammar from a set of example sentences tend to store a set of specific instances, rather than a generalized abstract knowledge of the process of grammatical combination. Further, recent models that do show limited systematicity do so at the expense of simultaneously storing explicit lexical knowledge, and also make use of both developmentally-implausible training data and biologically-implausible learning rules. Consequently, this research program develops a novel unsupervised neural network architecture, and applies this architecture to the problem of systematicity in language models.

In the first of several studies, a connectionist architecture capable of simultaneously storing explicit and separate representations of both conceptual and grammatical information is developed, where this architecture is a hybrid of both a self-organizing map and an intra-layer Hebbian associative network. Over the course of several studies, this architecture's capacity to acquire linguistic grammar is evaluated, where the architecture is progressively refined until it is capable of acquiring a benchmark grammar consisting of several difficult clausal sentence structures – though it must acquire this grammar at the level of grammatical category, rather than the lexical level.

The final study bridges the gap between the lexical and grammatical category levels, and develops an activation function based on a semantic feature co-occurrence metric. In concert with developmentally-plausible sensorimotor grounded conceptual representations, it is shown that a network using this metric is able to undertake a process of semantic bootstrapping, and successfully acquire separate explicit representations at the level of the concept, part-of-speech category, and grammatical sequence. This network demonstrates broadly systematic behaviour on a difficult test of systematicity, and extends its knowledge of grammar to novel sensorimotor-grounded words.

Acknowledgements

I can not imagine a student who has had the pleasure to work with a group of better, brighter, or more supportive people, and it is my sincere honour to be able to express my thanks to them and their efforts.

To my academic parents, Scott Watter and Karin Humphreys, I thank them for their unwaivering support, encouragement, generosity, and friendship over the past five years. Their selfless attention to my intellectual and professional development, their encouragement to explore science more generally, and their always open door (whether it be to talk about research, share funny stories, or help repair my heart from girl damage) makes me happy to call them my supervisors, and my friends. I simply know of no finer people.

To Lee Brooks, I thank him for periodically extending a wing that I might perch under when my research needed guidance (even when I wasn't aware that it did), that I might sit and listen to his wisdom for what always seemed too short a time. You are simply the brightest, most learned person I have ever known, and our discussions made me a better scientist, and taught me how to better think about the world. The day you came out of my committee meeting, shook my hand, and said "it sounds like you have some real exciting stuff here!" is extremely memorable for me. As you know, there is a saying among your students that you are ready to graduate when you can finally understand most of what Lee is saying, and maybe even add to it. I am glad I had the chance to experience this before you got sick, and left us. Out of sight, but not out of mind.

To Alex Sevigny, I thank him for serving as a member of my supervisory committee, for thoughtful discussions, and for teaching me more about language, especially in the first years of graduate school.

To my lab mates, I thank each of you for your friendship, uncountable amounts of laughter, thoughtful discussions, and just being yourselves. Maria, Sandra, Liz, Johanna, April, Molly, Melena, Juliana, and Ian (although particular thanks must be paid to April, for her legendary kung-fu skills, pirate accent, and yummy baking). One of my fondest memories as a graduate student is when the lab began to feel like a family, and I think of each of you as such.

To the concepts research group, and both Meredith and Sam in particular, I thank you for years of thoughtful and stimulating discussions. While our curiosity and love of research takes us on adventures across the world, our friendship thankfully extends across geographic borders.

To the many friends and family who have supported me over the years, for which there are too many to name here, I thank each of you for your encouragement, warmth, and kindness.

Last but not least, I have to thank my Dad, who taught me how to make, create, design, build, program, and experiment from a young age. Thank you for showing me the curiosity that you have about the world and how it works, for the ridiculous amounts of fun and learning experiences we have had over the years building all sorts of exciting things, and for being there for me, no matter what. You are an amazing dad.

Table of Contents

Chapter 1

Introduction

One of the most remarkable aspects of language is that infants are able to acquire the beginnings of a system capable of generating an astronomically large number of utterances in only their first few years of life. Even more remarkable is that infants are exposed to only a small fraction of the possible sentences they might generate, and yet from their limited experience they are able to progressively abstract the underlying components of language, including phonology, words, and grammatical knowledge, from a rich and diverse stream of language present in their environments. While we have collectively massed a great deal of data describing the progression of speech from the earliest of productions through to adulthood, the underlying process that allows each infant to undertake their journey along the path of language acquisition remains in large part a mystery. This thesis examines how infants acquire their earliest knowledge of language from the perspective of computational modeling and simulation. We examine the mental representations infants possess in order to support language, their representational structure, the processes that construct and make use of these representations, and the design of a computational-representational system capable of supporting these linguistic representations and processes.

1.1 Contexts of Language Processing and Representation in Computational Systems

Since the invention of physical computational hardware, a key avenue of discovery in understanding the processes involved in language use has been computational modeling. Abstractly, computational modeling is motivated by the notion that if we are able to arrive at a formal description of the function of an arbitrary system, either wholly or in part, we can use that formal description to construct a simulation of that system over time, and observe its interactions. If these interactions match our observations of how a given system behaves, we might be said to have an understanding of the underlying processes of that system, at least to the functional level that we have simulated. If our simulated observations disagree with those we observe in the physical system, we can re-evaluate and further develop our models until the two agree. Simulation has been broadly applied as test of understanding and a source of predictions in the physical science, and has been broadly used in areas such as fluid dynamics (e.g. Sharp, Long, and Khan, 1990), astronomical simulation (e.g. Ida and Makino, 1992), and electronic design (e.g. Bailey, Briner, and Chamberlain, 1994).

From a historical perspective, simulation in computational systems necessitated that whatever formal or mathematical description of a problem one arrived at, one would need to transduce this specification into processes the computer was capable of undertaking. While these atomic processes were first focused on numerical computation, the desire to represent and manipulate text soon lead to the development of character representations – if for nothing more than a mechanism to more easily represent the program of one's simulation at a higher-level than the atomic machine code of a computational system (e.g. McCarthy, 1960), which was laborious for a human to specify. Unsurprisingly, early computer scientists also recognized the ability of their new set of computational tools to specify and process information for tasks that, until then, had been purely the domain of humans, and a young group of computer scientists – John McCarthy, Claude Shannon, Nathan Rochester, and Marvin Minsky – organized the Dartmouth conference in 1959, which is commonly recognized as the birthplace of the field of artificial intelligence as it is studied today. Over the next few years and coming decades, the formal study of artificial intelligence enjoyed an enormous synergy with the study of computational languages for programming above the machine level, as well as both recent developments in understanding the underlying structure of human language at the level of the sentence (Chomsky, 1957), and the emerging field of cognitive psychology that itself was beginning to construct "information processing" models of mental processes (Broadbent, 1958). These early cognitive models were no doubt themselves inspired by the Von Neumann architecture (von Neumann, 1945, 1993), the pervasive computing architecture at the time. A central feature of this architecture is its modular nature, where it consists of a single large memory and a single general-purpose processing unit, interconnected with a relatively low-bandwith connection.

Information processing models continued to develop through the 1960s (e.g. Newell and Simon, 1972), where by the latter part of the decade, Collins and Quillian (1969; Quinlan, 1969) had developed and implemented a computational model capable of interpreting, learning from, and responding to statements of fact entered as text. This remarkable feat was enabled by implementing a system that was not only capable of undertaking some mechanistic process on the representations of text the computer was accepting, but also critically by demonstrating that a knowledge of the conceptual content of the words themselves was required. The structure of this "semantic memory" also allowed Collins and Quillian's computational model to exhibit some of the behavioural effects observed by cognitive psychologists in studies of human semantic memory, including the discrepancy in verification times for properties of concepts that seemed particularly central to their description (such as a *canary* having the property of *being yellow*), versus properties of those concepts that were less central still valid (such as *canary – breathes air*) and that took much longer to verify. A hierarchical model of semantic memory explained these effects, and the implementation of Collins and Quillian (1969) sparked a wide-spread interest into structures for knowledge representation that could also be computationally specified.

One of the most influential knowledge representation schemes to arrive in the next few years was the idea of representing knowledge in terms of *frames*, specified by Minsky in 1974.

Frame representations extended on the hierarchical specification of concepts in semantic memory, and placed particular focus on the structure of individual representations, as well as the overarching structure of all of semantic memory. Where concepts in the Collins and Quillian (1969) model had been specified at a relatively informal level (as in the case of birds having the properties of "can fly" and "have beaks"), there was a renewed interest in continuing the formalization of memory models toward the level of the concept. Minsky (1975) proposed the idea that concepts can be thought of as "frames" that contain a number of modifiers or role-bindings called "slots", where one could specify the typical agents, preconditions, or effects of a given concept or situation – where the content that filled these slots were themselves links to other frames. For example, one's frame of "car repair" might have the slots "tools required" and "preconditions", which might be filled by the values "ratchet" and "a knowledge of auto mechanics", respectively. These values are themselves pointers to other frames that describe the concepts of ratchets and auto mechanics.

Minsky's work on frames was influenced by both Schank's (1972) early work on specifying knowledge structures, as well as the progressive development of the concept of the schema (which later evolved into the script, Schank and Abelson, 1977) – both less formal specifications of an event-structure than frames. Schank (1972) was also concerned with the formalization of knowledge structures that could be used for text processing, and proposed an underlying theory of conceptual processing based on a propositional representation scheme called conceptual dependency theory. Representation schemes based on propositional logic began to gain ground, as they provided a framework for symbolic computation to take place on textual information, and merge the seemingly high-level representations of language with a mechanism for inference that was easily described in terms of a series of logical operations (e.g. Deliyanni and Kowalski, 1979).

It is around this same time that the approaches to artificial intelligence and cognitive modelling began to, in a small way, depart from one another – or at least, a divide begins to form separating representational and inference methods based on the formal specification of first order predicate logic, and those based on the as-yet much less formal network storage model methods of semantic memory (Collins and Loftus, 1975). The formalization of network models was to become a top priority, and extensive efforts were made to provide a unifying framework for expressing semantic networks that was also amenable to the processing and inference methods that first order logics were demonstrating. Particular effort was paid to the specification of links between the concepts in network models (such as is-a, or kind-of; Brachman, 1977, 1979; Woods, 1975), which up until that time had formalisms that greatly varied between researchers. Brachman also critically identified that network models and their links were describing knowledge across a variety of different *levels of representation*, from the linguistic level (which specifies knowledge at a high level), through to the conceptual level (where concepts are specified, e.g. the conceptual dependency theory of Schank, 1972), and down to the epistemological level (where formalisms supporting the specification of concepts are supplied, such as the inheretence of Collins and Qullian's model, 1969). The epistemological level is ultimately supported upon the logical level and implementational levels, which supply basic propositional logic and atomic computational or representational primities upon which the epistemological level can be constructed. This not only formalized the specification and simulation of semantic network models, but also layed the foundation for what became the study of ontological engineering (see, for example the work of Gruber, 1993; Miller, 1990; Vossen, Diez-Orzas, Peters, 1997).

1.2 Connectionism

In addition to his work on specifying formalisms for knowledge representation, Minsky was also interested in processing systems that functioned at a much lower level – the level of the neuron. Neural models were not new, where many of the semantic network models of memory were likely inspired by the recent discovery of the associative properties of linked neurons (Hebb, 1949), although semantic network models were specified at a much higher level. Minsky (1954) had much earlier laid the ground work for simulating neural systems in computational hardware, and in 1972 with Seymour Papert (Minsky and Papert, 1969) gave a characterization of the capacities of learning in layered systems of a computational abstraction of the neuron, called a Perceptron (Rosenblatt, 1958).

Neural network models offered a fundamentally different computational approach to cognitive modeling. Where models of the time tended to be derivatives of an information processing model of cognition, and resembled the Von-Neumann architecture of the computational systems available at the time in that they the processed information serially, neural network models described vast numbers of interconnected processors, each capable of performing simple computations, learning, and communicating with one another. Neural models also offered a measure of biological plausibility to cognitive models, resembling the biological structure of brains far more than serial models.

Connectionist models, as they came to be known, also offered a natural computational vehicle to explore network models of language and semantic memory, as the computational implementation of a network model could be a near direct replica of the graphical description of semantic network, and in this way connectionist models appeared much more high-fidelity representations of cognitive processes. Specialized architectures began to spring up in order to simulate massively-parallel models of semantic networks and their connectionist equivalents (e.g. Hillis, 1989), and the critical formulation of learning rules that enabled connectionist systems to be trained (Rumelhart, McClelland, and the PDP Research Group, 1986) lead to a number of demonstrations of connectionist models performing cognitive functions across a variety of tasks, including language (e.g. visual word recognition: McClelland and Rummelhart, 1981; sentence comprehension: McClelland, St. John, and Taraban, 1989, learning past tense verbs: Rumelhart and McClelland, 1986). So called "localist" representation schemes that ascribed each neuron (or "node") in a connectionist model to a labeled concept in the directed graph specification of a network model of semantic memory also gave way to representation schemes that "distributed" the representation of a concept across an entire layer of nodes in the network, which were found to account for the graceful degradation of language and representational function observed in patients with brain damage. As cognitive models of language, connectionism was beginning to provide broad and robust accounts of language learning, function, and degradation.

1.3 Systematicity in language processing

One of the criticisms of connectionist models as viable models of cognition came from Fodor and Pylyshyn (1988). Much like Woods (1975) had identified over a decade earlier for network models, Fodor and Pylyshyn argued that connectionist models were often specified at different *levels of representation*, and that one of the key levels they were now being specified at was at the level of symbol manipulation and cognitive architecture, which was inappropriate for a model based on low-level neural functioning. While from a reductionist perspective brains certainly implement cognitive systems, and as such cognition can ultimately be grounded in neural systems, current connectionist models of language were using this low-level formalism to explain too high a level of functioning, and ignoring critical aspects of the problem of language. Fodor and Pylysyn (1988) argued that connectionist models failed to exhibit *systematicity* in language processing, which is the property of being able to generalize or extrapolate from a set of numerous specific instances into a set of few general properties or rules. Where connectionist models could successfully process the exact sentences they had seen before, they were unable to successfully demonstrate processing even minor but untrained variations of those sentences, or generalizing to novel sentences or words. Further, even if a single demonstration of systematicity in a connectionist system was developed, this would be insufficient – systematicity is a *general* property of cognition (Fodor and McLaughlin, 1990), and should too be a broad property of any model of cognition. Connectionism was attempting to operate at the level of the symbolic processing of a cognitive architecture, and an artifact of such a low-level architecture attempting to account for symbolic behaviour was producing models that failed to account for key aspects of cognitive architecture, like systematicity.

Because the arguments of Fodor and Pylyshyn (1988) were centrally aimed at models that made use of the connectionist architecture described by Rummelhart and McClelland (1986) – namely the three-layer feed forward architecture and the backpropagation of error learning algorithm (although Fodor and Pylyshyn were careful to note that their argument subsumes all of connectionism, not just a single architecture) – others attempted to demonstrate systematicity across different network architectures. Because any such demonstrations were likely to be partial, and to help define a metric for measuring systematicity in connectionist systems, Hadley (1994) teased apart the idea of systematicity into what he called *weakly* and *strongly* systematic behaviour, each of which are tied to specific demonstrations a network must make before it meets either criterion. A network that is said to be weakly systematic must be able to not only acquire representations of a set of trained sentences, but must also be able to accept novel sentences composed piecewise of words that occured across trained sentences, with the caveat that these words occur at the same location in the novel sentence as they were presented in the familiar. For example:

1. the cat ran
2. a mouse slept
3. a cat slept

Given training on the sentences "the cat ran" and "a mouse slept", the network must demonstrate the capability of also accepting the sentence "a cat slept", where this novel sentence is composed of words from familiar sentences, in familiar sequential locations. Because "cat" occurred as the second word in sentence 1, the network should also be able to generalize and accept it as a valid substitution for the second word in sentence 2.

This is the heart of the matter – in terms of language processing, systematicity is about being sensitive to the underlying syntactic or grammatical structure of a sentence. Further, this knowledge of the underlying grammatical structure must come about not through previous exposure to a given sentence, but through abstracting both the general combinatorial properties of words, as well as the valid set of combinatorial rules that define grammatical combinations of words, from only a small and limited subset of the total possible utterances of a given language.

In this way, the demonstration for strong systematicity requires that a network be able to acquire a knowledge of word category (noun, verb, adjective, etc), as well as a knowledge of a valid set of sequences of word category (sentence structures), and to successfully demonstrate accepting valid novel sentences composed of familiar words and sentence structures across syntactic position. For example:

4. the mouse slept
5. the cat chased the mouse
6. the mouse chased the cat

A system exhibiting strong systematicity must accept the novel sentence "the mouse chased the cat" as valid and grammatical, given previous exposure to the sentences "the mouse slept", and "the cat chased the mouse". This requires that the network be able to ascribe a grammatical category to each word in the sentence (for instance, "noun", in the case of "cat" and "mouse"), as well as to parse each sentence structure into its component grammatical categories, and accept novel combinations of words in familiar sentence structures that do not violate grammatical category boundaries. Hadley (1994) further specified that for a network to be considered strongly systematic it must also be able to acquire grammatical category across sentence structures that include embedded clauses, such as "the dog saw the cat who chased the mouse" (a right-branching clause), or "the cat who the dog saw chased the mouse " (a centre-embedded clause).

Hadley (1994) noted that while some connectionist models had displayed weak systematicity, there had as yet been no convincing demonstrations of strong systematicity. To demonstrate that connectionist models could indeed display strong systematicity, Hadley and Hayward (1997) constructed a connectionist model based on Hebbian-competitive learning, although critically, the model also contained hybrid elements that implemented a semantic role-binding system with well-defined links between concepts, similar to Minsky's (1975) specification of Frames. While Hadley demonstrated that the model exhibited strong systematicity, it was not itself viewed as a successful demonstration of systematicity in a trained connectionist architecture. Fodor and Pylyshyn (1988) admitted that some organization of neural systems must exhibit systematicity because brains too exhibit systematicity – their argument was not that neural systems were in principle incapable of implementing symbol systems, but rather that they had not yet demonstrated a general capacity to be trained into symbol systems.

1.4 Contemporary Approaches to Modeling Systematicity

The recurrent neural network architecture of Elman (1990) extended the three-layer feed-forward architecture described by Rumelhart and McClelland (1986) with facilities that allowed information to flow through a network not simply layer-by-layer in a stage like fashion, as in the case of feed-forward networks, but also to flow backward in the architecture. Elman described a procedure whereby recurrent connections in the hidden layer of a three-layer feed-forward network could be used to supply the hidden layer with information processed in a previous time-step, and provide the network with a "temporal context" of processing. This had dramatic implications for sequence processing in connectionist systems, and greatly increased their utility in modeling language beyond the level of the single word, and into the level of sequences of words, or the sentence. The simple recurrent network (SRN) became a standard tool for language modeling, and demonstrations included models of grammatical structure (Elman, 1991) and the lexicon (Elman, 1995), to name only a few.

The ability for the SRN to display systematicity was evaluated by Marcas (1998), who demonstrated that the networks Elman was using to simulate the acquisition of grammar learning were unable to generalize beyond their training set, even for simple tasks. Marcas trained a SRN to learn sentences of the form "a X is a Y", where X and Y represented identical instances of a noun, such as *house*. While the network performed well on previously trained instances, such as "a cat is a cat" or "a train is a train", the network was unable to respond correctly to novel sentences such as "a blicket is a Y", where *blicket* was a previously untrained noun. This pattern of results is consistent with a network that is acquiring specific instances, rather than general processes, and was echoed by Prasada and Pinker's (1993) evaluation of earlier connectionist models of English past-tense based on the Rumelhart et al. (1986) architecture.

Similarly, Van der Velde et al. (2004) evaluated the potential for simple recurrent networks to display strong systematicity, and concluded that the networks were generally incapable of demonstrating even weak systematicity. Van der Velde et al. trained a simple recurrent network on what became a benchmark language dataset for evaluating systematicity in connectionist systems. Their data set includes three sentence types, including simple (N – V – N, where "N" represents "noun" and "V" represents "verb), right-branching (N – V – N – Who – V – N), and centre-embedded (N – Who – N – V – V - N) clausal sentence structures, as well as a small vocabulary of 8 nouns and 8 verbs. The SRN was trained on a small subset of these possible sentences, and failed to produce widely systematic behaviour.

Frank (2006) suggested that the failure for Van der Velde et al. (2004) to produce systematicity in the SRN may have been a function of the properties of the simulation, rather than a general property of the network architecture. Specifically, Frank (2006) suggested that a relatively low number of nodes in the hidden layer of the SRN may have impaired the network's capacity to generalize. Further, the size of the vocabulary used by Van der Velde et al. was particularly small – where infants tend to receive a great variety of language input – and as such, the network may have been unable to generalize. In response to these issues, the network may have been subject to a process of overfitting, where it is capable of storing each individual word, and does not require generalization to effectively store each trained pattern, or similarly subject to a process of underfitting, where the network simply has too few resources available to support the capacity for generalization.

Frank (2006) proposed a number of further simulations, using both the simple recurrent network, as well as a related variant, the Echo State Network (ESN; Jaeger, 2003). Where the recurrent connections between the hidden and input layers of the SRN are complete (and potentially available for training), the corresponding connections of an Echo State network are sparse, lossy, and of relatively low weight. In this way, where the SRN must learn a process of "dimensionality reduction" to generalize grammatical category from a broad set of instances, the ESN has a mechanism of dimensionality reduction built-in at the level of the connection, and as such, may be more amenable to displaying systematic behaviour. Both networks were trained on the benchmark training set of Van der Velde et at. (2004), consisting of simple, right-branching, and centre-embedded clausal structures, with a variably-sized and vastly expanded vocabulary of nouns and verbs. The networks were trained on up to 25,000 utterances, a small subset (approximately 0.4%) of the total unique sentences combinatorially producible given the vocabulary size.

In analyzing network performance, Frank (2006) argued that van der Velde et al. may have used too binary a measure of systematicity, and that even if a network did not display systematic performance in general, it may still exhibit performance above a level reflective of completely unsystematic behaviour. Frank introduced a performance measure, later referred to as FGP, that compared the performance of the network to a bigram model of grammatical prediction, which it used as a baseline measure. FGP is defined on the real interval (-1:1), such that a network that is behaving completely systematically would produce an FGP value of 1, a network behaving only as a bigram model would produce an FGP of 0, while an FGP of -1 is representative of a network that produces entirely ungrammatical predictions.

Both the SRN and ESN models were evaluated across a variety of network parameters and lexicon sizes, and produced, on average, FGP scores indicative of strongly systematic performance, with the ESN model generally performing better than the SRN model in measures of systematicity, although substantial dips in the performance of both models in grammatically difficult situations were evident. This was taken as support that these recurrent connectionist architectures, while not demonstrating a general property strong systematicity, are exhibiting behaviour that is substantially above the level of behaving entirely unsystematic, and at a level where at least some concept of generalization is present. Brackel and Frank (2009) demonstrated that this aspect of systematicity is a general property of SRNs across a variety of training circumstances, and that the internal representations of the network are likely learning to produce similar representational states for words of similar grammatical category, where the network can use this general process of transducing words into grammatical categories as a mechanism to behave strongly systematically within and across sentence structures. This general property of SRNs across a variety of circumstances was taken as a demonstration contra Fodor and Pylyshyn (1988), in that connectionist architectures have the potential behave systematically, at least some of the time.

Extending upon this work, Farkas and Croker (2008) constructed an unsupervised model of systematicity in language processing, in an effort to extend the result of Frank (2006) by demonstrating systematic processing across very different connectionist architectures. Where the SRN and ESN models of Frank (2006) make use of the backpropagation of learning algorithm (Rumelhart and McClelland, 1986), a supervised error-driven learning algorithm that requires constant input from a trainer or "oracle" during training, Farkas and Croker (2008) used an unsupervised architecture based on the self-organizing map (Kohonen, 1982, 1995). The self-organizing map (SOM) is an architectural abstraction of the topographic organization observed in perceptual cortex, and combines unsupervised forms of both cooperative and competitive learning. Because Kohonen's (1995) specification for the self-organizing map describes an architecture that is particularly capable of representation and categorization, but almost completely incapable of processing with that information, Farkas and Crocker (2008) presented and made use of a derivative architecture, the RecSOMsard, that incorporates temporal sequence processing capabilities using methods similar to the Recursive SOM (RecSOM; Voegtlin, 2002) and the Sequential Activation Retention and Decay Network (SARDNET; James and Miikkulainen, 1995).

The RecSOMsard model was trained on the benchmark grammar of Van der Velde et al. (2004) using a similar vocabulary size and training proportion to the models of Frank (2006). The RecSOMsard model exhibited performance similar the ESN model, with a mean FGP of

0.85 across simple, right-branching, and centre-embedded sentence structures. While this mean performance was indicative of a network with a high probability of demonstrating strong systematicity, specific performance at each grammatical transition across all sentence parses fluctuated dramtically, with performance as low as 0.6 as grammatical prediction difficulty increased. Overall this pattern of performance was very similar to Frank's (2006) ESN model, and demonstrative that systematic behaviour can be observed across a variety of very different connectionist architectures, at least some of the time.

1.5 Representational Grounding in Language Learning

While not specifically addressing the issue of systematicity in connectionist systems, others have investigated whether the expressivity of the input set might influence performance in grammar learning. Jean Mandler has extensively studied the conceptual world of even very young infants, and describes (Mandler, 2004) the rich conceptual representations of pre-linguitic infants that have been uncovered across a variety of experimental paradigms, including habituating to familiar concepts, looking preferences, and differed imitation. To investigate whether the availability of conceptual information would influence grammar learning, Howell, Jankowicz, and Becker (2005) constructed a database of sensorimotor conceptual representations consisting of 352 nouns and 89 verbs across 97 and 84 developmentally-plausible perceptual and kinesthetic noun and verb feature dimensions that 8- to 28-month old infants are likely to be sensitive to, such as "is red" or "involves mouth motion". Howell et al. (2005) showed that while a SRN was capable of acquiring a simple grammar of N – V – N sentences using only abstract symbolic lexical representations, sensorimotor grounded conceptual representations greatly improved performance, suggesting that the affordances the world has to offer may help guide the infant in early grammar learning.

Broadly, representational grounding (or symbol grounding; Harnad, 1990) refers to the problem that the representations used in computational models or connectionist simulations are closer to the kinds of symbolic representations in books rather than brains. That is, Harnad (1990) explains, there is nothing about the word "book" itself that conveys any information about the concept of a book, such as that it has pages or contains information, but rather the word "book" itself is purely symbolic, and alone without its conceptual referent is equivalent to a bare number devoid of semantic content. Searle (1980) suggested that even if a computational system appears to carry out complex, intelligent, and cognitive behaviour (such as answering questions in Chinese), the mechanisms internally used by that computational system are critical as to whether the system could be said to have a grounded understanding of those concepts, or whether (on the other end of the spectrum) the system is performing a completely ungrounded symbolic look-up operation and has no understanding of the meaning of its representations or behaviour. Harnad (1990) extended this thought experiment, comparing ungrounded symbols in computation as equivalent to the task of trying to learn Chinese using only a Chinese-to-Chinese dictionary.

It is often argued that a critical aspect of grounding is the ability to have direct access to input from and the potential to effect changes to the world, a perspective known as embodiment (e.g. Clarke, 2008). The concept of embodied cognition is pervasive across the study of intelligent systems, from approaches that propose extreme levels of embodiment with virtually no internal representation in cognitive robotics (e.g. Brooks, 1991), to others that suggest that most aspects of thought and representation occur not centrally in either the mind or the world, but across a continual exchange between the two (e.g. the extended mind hypothesis, Clarke, 2008). In either case, it seems likely that the world supplies us with at least a vast amount of the information that we use to learn about it, and that computational models that are sensitive to grounded information can potentially capitalize on the benefits it has to offer.

1.6 Thesis Summary

One of the central open problems in developmental knowledge representation, described in detail by Mandler (2004), is that while we are beginning to understand the rich conceptual world of even very young infants, developmental theorists have a great deal of trouble "getting the child going" – that is, describing the system or mechanisms that acquire the very earliest of concepts, from which knowledge is thought to bloom. Computationally, this exact problem is at the very foundation of artificial neural network research, and much of the fundamental study of learning rules examines the theory and methods of how connectionist networks can acquire a system of representation. Through the evolution of unsupervised network architectures, the architects of neural systems have continually asked how they might construct an ever-more independent *Tabula Rasa* – a complete computational-representational substrate ever-more capable of acquiring representations by virtue of its experiences with the world. This journey has led the field through its initial success with supervised networks, to more recent explorations in the development of unsupervised network architectures. In terms of developmental knowledge representation, these latter developments are particularly synergetic – our motivations have transitioned from asking how one might scribe the world upon a blank slate, to asking what the slate *itself* might require in order to be sensitive to the world in computational and representationally interesting ways.

This thesis presents work broadly centered around the notion of developing an unsupervised connectionist architecture in the above context, with several goals. The first goal is to develop a model capable of displaying a general property of strong systematicity, such that systematic behaviour is observed across grammatical class and sentence structure acquisition, irrespective of relative difficulty. The second goal is to address the problem of representational grounding (Harnad, 1990), such that the network could be argued to possess grounded and developmentally-plausible conceptual representations that it makes use of to acquire its knowledge of the combinatorial structure of language. The third goal is to separate the often unattended-to boundary between representation and processing in connectionist systems, such that each level of processing in the model will contain explicit and separable representations of conceptual knowledge, grammatical category, or valid grammatical sentence structures. Supporting these first three goals, the fourth goal is to explore the further specification of the relatively high-level description of unsupervised self-organizing connectionist architectures for sequence learning, and to develop a system capable of supporting the first three goals while also describing low-level inter-node network dynamics and approach the ideal of a network architecture expressed as a cellular automation. In this way the overarching goal of this work is to explore the possibility of creating a limited symbol system that displays strongly systematic behaviour on the task of acquiring grammar through pre-linguistic lexical knowledge. Further, this symbol system is to be implemented at a relatively low level of neural specification, first connecting between Fodor and Pylyshyn's (1988) symbolic and connectionist levels of representation, then between connectionist levels and the level of embodied representations grounded in the world.

Chapter 2 begins to develop such a low-level network specification based on common approaches to extending the unsupervised self-organizing map architecture in an effort to acquire temporal sequence information. Chapter 3 examines a multi-layer version of this architecture for an abstracted grammatical learning task that does not require the use of systematicity, and examines the high-order topographic organization of these representations. Chapter 4 experimentally examines the capabilities of the SRN to acquire abstract and deeply ambiguous grammars, and concludes that the proposed architecture needs to migrate from a mechanism of feed-forward processing to recurrence in order to obtain a similar level of performance. Chapter 5 implements this recurrent architecture, and compares its performance on a restricted version of the Van der Velde et al. (2004) benchmark grammar that does not require systematicity. Chapter 6 alters the architecture, and develops a method for grammatical class abstraction based on semantic co-occurrence while testing performance on a hard and deeply systematic version of the Van der Velde et al. benchmark set. Chapter 7 discusses the ultimate performance of the model, comparing it to past approaches, and assesses the general implications.

Chapter 2

Chimaera Networks for Self-Organizing Grammar Acquisition

2.1 Introduction

The self-organizing map, or SOM, was originally formulated by Kohonen (1982) as an unsupervised analog to the topographic organization found in perceptual cortex (e.g. Cansino et al., 1994). The self-organizing map typically consists of a single-layer *n*-dimensional spatial array of nodes, each of which is capable of representing a *k*-dimensional data vector. Representing data vectors in a SOM is explicit, both in that each node represents a given vector, and that the learning rule explicitly modifies a subset of these nodes' representations to more closely represent a given input vector. In this way a single-layer SOM can be thought of both as a substrate for data-vector representation and classification, yet without extensions (such as recurrent or multilayered SOMs), is unable to represent associative processes or temporal sequences. The SOM architecture has been applied to a diverse set of phenomena, including grammatical and semantic feature classification (Howell et al., 2005; Ritter and Kohonen, 1989), acquiring phonetic categories (Behnke and Whittenburg, 1996), and visual contour integration (Choe and Miikkulainen, 2004).

Since its development, the SOM has experienced a proliferation of architectural extensions that add the ability to represent temporal sequences of data through a variety of mechanisms (see Mozer, 1994, for a review of approaches to temporal representation). One of the first such architectures, the Temporal Kohonen Map or TKM (Chappell and Taylor, 1993), was designed to make a simple, biologically plausible addition to the original Kohonen map that allowed temporal context to be included in the network's topography, without resorting to explicitly supplying the network with a fixed temporal context window (e.g. Kohonen, 1991). The TKM accomplished this through the use of leaky integrators, which allowed the learning rule to factor recently winning nodes into subsequent best-matching node calculations.

Many contemporary approaches to temporal representation, such as the RecSOM (Voegtlin, 2002), SOM for Structured Data (SOMSD; Hagenbuchner, Sperduti, and Tsoi, 2003), and Merge SOM (Strickert and Hammer 2003), use a scheme where the data-vector representation of each element in a sequence is a combination of 1) the features that make up that representation (e.g. the RGB values, in the case of a colour), and 2) the "context" or position of that data vector in a sequence, which usually takes the form of recurrent connections that supply snapshots of the network's processing at previous time steps. While this processing mechanism has seen a great deal of success, from an epistemological standpoint, this representation scheme would seem to suggest that individual instances of a given "concept" are repeatedly stored in a slightly different form each time they appear in a different sequence of events (the TKM is also not immune; see Chappell and Taylor, 1993, sentence simulation). Aside from the issues this places on cognitive economy – i.e. having multiple copies of concepts stored each time we see or use them in a different sequence – it also suggests that there is something fundamentally different about the representation of the concept of "blue" when it appears in the sequence "red-orange-blue-green" rather than "desk-blue-road-leaf" – or, even more profoundly, something fundamentally different about the representation of "blue" in the sequences "red-orange-blue-green" and "green-blue-red-orange", simply because its position in the sequence has changed. There is a wealth of evidence that suggests that feature information for a given concept, and temporal sequence information are very different kinds of knowledge that likely rely on different processing mechanisms (Ullman, 2001), and this has enabled considerably more complex psycholinguistic models of morphosyntactic use (Chang, Dell, and Bock, 2006) with dual-representational models based on simple recurrent networks (Elman, 1990). In this way, storying multiple copies of a given concept or representation in multiple temporal positions is not a theoretical claim, but rather has been used as a method to make the temporal computation much easier in a connectionist system. Even explicitly temporal architectures with complex temporal dynamics such as SARDNET (James and Miikkulainen, 1995) make use of similar high-level or supervised mechanisms that functionally represent temporal context using multiple copies of a given concept across different temporal contexts. Because this multiple-temporal-copies scheme does not scale well computationally or representationally, and sidesteps the problem of having to construct a symbol system capable of the abstract reuse of concepts across arbitrary contexts, it would be advantageous if this limitation could be overcome. While storing multiple copies of representations has reduced the complexity of implementing temporal models, it is our responsibility to ask whether storing this information together makes sense, or whether different types of knowledge and different processing mechanisms necessitate a different type of system capable of simultaneously storing and processing these two types of knowledge.

2.1.1 Formal Description

Taking a step back from our representational characterization of self-organizing map architectures, the SOM is typically described not as a mechanism for storing representations, but rather a mechanism for dimensionality reduction and spatial clustering. As such, the SOM is typically defined not in terms of a computational abstraction of nodes that each store a data vector representation, but rather as a set of nodes that each have a vector of weights that connect them to some input vector representation. While these two abstractions are functionally identical and produce the same behaviour, the representational characterization of the SOM as a method of storing data vectors rather than a system of weight modification will be used here – both because we are specifically interested in an architecture that is capable of explicitly representing arbitrary data vector representations of conceptual information, and also because we will be using the term "weight" to refer to the temporal associations *between* these representations later in our formulation of an architecture with separate representations of data vectors and their temporal associations.

That being said, the SOM (Kohonen, 1990) is typically described as a set of nodes ***i***, where 1 ≤ ***i*** ≤ N, each of which has a corresponding weight vector $\boldsymbol{w_i}$ between that node and an input vector ***x(t)***. At a given epoch ***t***, the weights of the network are typically modified based on a given node's spatial proximity to a "best-matching" node of index ***k***, defined as the node that minimizes the quantization error $\boldsymbol{E_i}$ between the input and weight vectors, typically evaluating them in terms of the Euclidian distance $\|\cdot\|$ such that:

$$k = \arg\min_{i\in(1,2,\ldots,N)} E_i \tag{1}$$

$$E_i = \|x(t) - w_i\| \tag{2}$$

$$\Delta w_i = \gamma h_{ik}\left(x(t) - w_i\right) \tag{3}$$

where γ is a learning rate 0 ≤ ***i*** ≤ 1 that linearly decreases as ***t*** increases, and $\boldsymbol{h_{ik}}$ is a "neighborhood" function, typically a Gaussian, that decreases as the spatial distance ***d(i,k)*** between nodes ***i*** and ***k*** increases on some metric, typically a 2D Euclidian or hexagonal lattice:

$$h_{ik} = e^{-\frac{d(i,k)^2}{\sigma^2}} \tag{4}$$

where the "neighborhood width" or radius-of-learning σ decreases from an initially large value (typically on the order of the size of the network on a spatial dimension) to 0 as ***t*** increases.

Approaches to temporal representation typically extend the original SOM specification using a mechanism of supplying temporal context from previous epochs, often by extending the quantization error $\boldsymbol{E_i}$ of the best-matching unit function to be sensitive to temporal information supplied by leaky-integrators or explicit recurrent context input. For example, TKM makes use of leaky integrators, such that the quantization error $\boldsymbol{E_i(t)}$ depends on activation not only from the current epoch $\boldsymbol{t}$, but also from previous epochs:

$$E_i(t) = (1-\alpha)\cdot\|x(t) - w_i\| + \alpha \cdot d_{t-1}(i,k) \tag{5}$$

where α, such that $0 \leq \alpha \leq 1$, determines the relative contribution of the current input vector and the temporal context to the activation of a given unit.

Instead of using leaky integrators, architectures such as RecSOM and SOMSD have an explicit representation of temporal context built into a separate subsection of the weights of the network, denoted $\boldsymbol{c_i}$, such that following (3):

$$\Delta c_i = \gamma h_{ik}\left(y(t-1) - c_i\right) \tag{6}$$

and where the output $\boldsymbol{y(t)}$ of a node $\boldsymbol{i}$ is computed as an exponential of the distance function:

$$y_i(t) = \exp(-d_i(t)) . \tag{7}$$

$\boldsymbol{E_i(t)}$ is similarly modified as in (5) to be dependent on both a match to the current input vector, as well as this explicit representation of temporal context:

$$E_i(t) = \alpha \cdot \|x(t) - w_i\| + \beta \cdot \|y(t-1) - c_i\| \tag{8}$$

where α, β > 0 are static parameters that determine the relative contribution of the current input vector and temporal context to the activation of a given unit. Because the computation time of an algorithm using a context vector consisting of the activation of each node in a layer can become large, particularly for large network sizes, some recurrent methods including SOMSD limit this context to include only the spatial index of the best matching node from the previous epoch i_{t-1}:

$$E_i(t) = \alpha \cdot \|x(t) - w_i\| + \beta \cdot d(i_{t-1}, c_i) \tag{9}$$

The remainder of this chapter develops a test framework to explore an alternative approach to simultaneously storing and processing conceptual information and sequence information. Section 2 describes this novel test framework, called a Chimaera SOM, in detail, and explores an approach that separates temporal information from the quantization error E_i. Instead, this architecture blends the concept of activation and Hebbian activation flow over time with a traditional SOM, and results in a self-organizing map hybrid with two representational systems capable of explicitly and separately storing both feature representations, as well as simple unique sequences of those representations. This sequence representation mechanism does not generalize to all cases, and it generates ambiguous predictions when the same concept is used across multiple sequences. Section 3 shows how this problem can be overcome through representing higher-order "transitions between concepts", as opposed to simply concepts themselves, in a multilayer Chimaera SOM network. This method has the benefit of retaining the dual representational system structure, while automatically generating transitional information for higher-layers through input from lower-layers. Section 4 describes some remaining limitations of this framework, and suggests possible resolutions.

2.2 Simple Sequences and the Chimaera Framework

Kohonen's (1982) specification for the self-organizing map describes an architectural abstraction – a neural system that describes behaviour not at the single-neuron level, but at a somewhat higher level, with an overarching learning rule inspired by Hebbian learning over an entire cortical area. It is this method of learning that gives rise to the "cortical column" topology of a SOM, where similar data vectors are represented in topologically close areas of the spatial network. It then seems natural to extend this architecture, originally inspired by Hebbian learning, with facilities to allow a form of Hebbian associative learning to simultaneously take place on larger scales as well.

2.2.1 An Activation Map, and Inter-node Hebbian Association

Recall that Hebbian learning (Hebb, 1949) defines a change in the weight between two nodes in a network as a product of the activation of those two nodes at a given time, $\Delta w_{ij}(t) = y_i(t) \cdot y_j(t)$. The SOM learning rule is relatively unusual among neural network architectures in that it often tends to be described (in our representational abstraction of the SOM) not in terms of the activation of nodes or of weights between nodes, but rather in terms of the magnitude of the difference between an input data vector and a given node's data vector. In a given epoch, the data vectors of a subset of the network's nodes are modified based on each node's spatial proximity to a node in the network that most closely resembles the input vector – not in terms of connection weights between nodes in the network, which the SOM architecture has abstracted away. In this sense, without a mechanism to supply explicit activation values for each node in the network, the abstractions of the self-organizing map and the low-level definition of Hebbian learning may at first seem incompatible.

Similar to RecSOM and other temporal SOM architectures, the Chimaera architecture extends the self-organizing map by introducing facilities for computing both the activation of a given node, but also adds a mechanism for computing associative weights between nodes. In this way separate systems exist for representing data vectors and temporal representations – the best-matching unit function (2) is independent of temporal processing, which is supplied by a completely separate Hebbian association network. At each epoch a full activation map for the network is computed, which is then used by a Hebbian-inspired rule to compute an association weight *from* each node, *to* each node in the network.

In a given epoch, the SOM learning rule typically searches for the best-matching node in the network based on the principle of minimizing the Euclidian distance between an input vector and a given node's data vector. In a similar way, we can use the same principle for computing an activation map. At each epoch, the Euclidian distance between the input vector and each node's data vector is calculated, much in the same way as is used in the SOM learning rule; however this distance is then directly used, in concert with activation that flows from other nodes, to compute the activation for a given node caused by the input vector:

$$y_i(t) = \frac{(1-\tau)}{\sqrt{k}} \cdot \left\| x(t) - w_i \right\| \qquad (10)$$

where $\boldsymbol{k}$ is the dimensionality of vectors $\boldsymbol{x(t)}$ and $\boldsymbol{w_i}$, and $0 \leq \boldsymbol{\tau} \leq 1$ is a scaling factor that produces activation $\boldsymbol{y_i(t)}$ on the interval (0,1) for vectors that are within the scalar proportion $\boldsymbol{\tau}$ of each other. For example, where $\boldsymbol{\tau} = 0.10$, only vectors whose Euclidian distance is within the closest 10% of the maximum distance between those two vectors will produce activation, scaled such that two vectors 10% different would produce an activation of 0, while a 5% difference would produce an activation of 0.5, and similarly zero difference would produce an activation of 1.0 . In this way a variable notion of similarity is present in the activation function, and is modulated by the value of $\boldsymbol{\tau}$. From an epistemological perspective this makes sense – we can assume it is unlikely that activating our representation of, say, apple, also significant activates our representation of dolomite, although they may share a few highly abstract features such as being naturally occurring. Values of $\boldsymbol{y_i(t)}$ that do not fall on the interval (0,1) are clipped to zero.

The Chimaera architecture includes a full association map describing asymmetric connection weights from each node, to each node in the network. These weights allow a certain portion of the activation $y_i(t)$ at a given node to flow to all the nodes it is connected to, and increase the activation of those nodes. Where a_{ji} represents the associative weight from node j to node i, the flow of activation f to node i from node j is defined as:

$$f_{ij}(t) = y_j(t) \cdot a_{ji}(t) \tag{11}$$

The total flow f to node i is the sum of all activation flowing to that node from each node j modulaled by the weights between i and j for each j:

$$f_i(t) = \sum_{j=0}^{N} y_j(t) \cdot a_{ji}(t) \tag{12}$$

Finally, the activation of a given node in the Chimaera network is dependent not only on the similarity of input and data vectors $s_i(t)$, or the flow of activation from other nodes $f_i(t)$, but also leaky-integrator activation left over from the previous epoch:

$$y_i(t) = y_i(t) + f_i(t) + \left(y_i(t-1) - k_d\right) \tag{13}$$

where k_d is a constant activation decay parameter, each contribution $y_i(t)$, $f_i(t)$, and $y_i(t-1) - k_d$ is only valid on the interval (0,1).

2.2.2 Representing Temporal Sequences with Hebbian Learning and Temporal Decay

At each epoch, the associative weights between nodes are modified based on a Hebbian-inspired rule. This rule is essentially Hebbian-learning, but includes two scaling factors to mediate the modification of weights over time, and is only used if the activation of both nodes exceeds a minimum 'noise' threshold:

$$\Delta a_{ji}(t) = k_m \cdot k_s(j,t) \cdot y_i(t) \cdot y_j(t) \tag{14}$$

Additionally, to prevent the cascading multiplicative transmission of activation, if the sum of all association weights from a given node ***i*** to all nodes ***j*** is greater than 1, these weights are linearly scaled such that the sum is equal to one.

The k_m parameter is constant across the network, and serves as a global scaling factor to prevent rapid changes in the network. In a sense, this factor mediates how quickly the association weights change, and as a result, how quickly the network learns associations. The k_s parameter creates a unidirectional association dynamic, and exhibits a selective increase in the magnitude of modification of weights from nodes that have *recently* been highly activated to other activated nodes. This parameter is set to zero when the activation of a given node in the *previous* epoch was less than some threshold value k_{s_thresh}, and one otherwise:

$$\begin{aligned} k_s(j,t) &= 1, \quad for\ a_j(t-1) \geq k_{s\_thresh} \\ k_s(j,t) &= 0, \quad for\ a_j(t-1) < k_{s\_thresh} \end{aligned} \tag{15}$$

Temporal sequences are represented in Chimaera networks as a consequence of the Hebbian association process acting over time. As a series of data vectors are serially presented to the network, specific nodes corresponding to those data vectors will highly activate. When the t^{th} input vector in a sequence is presented, the activation caused by the presentation of the $(t\text{-}1)^{th}$ input vector will progressively decay to zero at a rate mediated by the k_d decay parameter. Similarly, specific nodes corresponding to data vectors representing the t^{th} input vector will activate. As such, at any given point in a sequence, the region representing the t^{th} input vector will be highly activated, where regions representing the $(t\text{-}m)^{th}$ input vectors, where *m* is some positive integer, will have significantly less activation as *m* increases (i.e. the earlier an input vector is in a sequence, the less activation the region representing it will have at some arbitrary point in that sequence). This decay profile is similar to other leaky-integrator activation extensions of the SOM, including SARDNET (James and Miikkulainen, 1995).

This simultaneous activation of multiple network regions corresponding to temporally-close portions of the input vector sequence allows the Hebbian learning rule to associate these regions. Specifically, the weights between nodes corresponding to input vectors *(t-1)* and *t* will associate. As a result of the k_s parameter, described above, the weights in the temporal direction from the $(t\text{-}1)^{th}$ input vector to the t^{th} input vector will associate to a much higher degree, as a consequence of the nodes for the $(t\text{-}1)^{th}$ input vector having just recently been highly activated. Over time, as the weights between regions preferentially increase in the direction of positive *t*, the flow of activation between nodes will cause activating the t^{th} nodes to also significantly activate the $(t+1)^{th}$ nodes. This form of prediction-over-time constitutes the Chimaera's temporal sequence learning, and is similar in form to a Synfire chain (Herrmann et al., 1995).

2.3 Single-layer Simulations

2.3.1 Visualizations

A code-base implementing the Chimera network architecture in parallelized OpenMPI C++, as well as several data visualization techniques for the output of this code base, was developed. Two techniques are typically used when visualizing SOM data: directly viewing the data vectors, and constructing a similarity map. Directly viewing the data vectors of a map is often difficult, as these vectors can be of large dimensionality. In cases with large vectors, automated or hand-tagging the best-matching unit for a given input vector can be used, such that one produces a labeled graph containing labeled best-matching locations for each input vector.

Adapting this technique, in cases where we can appropriately describe the input vectors in 3 or fewer dimensions, we can map each dimension of the SOM's data vectors to a colour value (red, green, or blue), and in this way directly view the data vectors contained within each node in the map. Because this technique is largely only useful for input vectors with fewer than 4 feature dimensions, and the tagging technique supplies only information on the best-matching nodes in a network, the similarity map provides an alternate technique that both extends to higher dimensional data vectors, and gives a measure of data vector content for each node in the network. The similarity map visualization calculates how similar each node in the network is to all other nodes in some small region near that node, and represents this as a single intensity value scaled from very similar (zero) to very dissimilar (one). This method is particularly useful for determining the organization of regions of similar data vectors in a SOM, where one is able to view general clustering based on similarity.

Two additional visualization techniques were developed for the Chimaera architecture. The first visualizes the activation map, and in concert with the standard SOM visualization techniques allows one to examine the activation and decay of specific regions and data vectors within the Chimaera network over time. The second technique allows one to inspect the association weights between each node in the Chimaera network. For a 2D spatial array of nodes, a full association map between each node will contain four dimensions: two for the origin node, and two for the destination node. A discrete 4D space can be represented as a two dimensional (global) array of two dimensional (local) arrays, and the association weight visualization makes use of this by organizing the 4D association map into a 2D origin (global) array of 2D destinations (local). At each point in this 4D array, the association between an origin and destination node is represented by an intensity value, scaled from a very low weight (dark) to a very high weight (bright). To ease the interpretation of the association network, this visualization also outlines each local array with a colour representing the value of the data vector at that location in the network.

2.3.2 Simulation 2.1

Figures 2.1 and 2.2 show a single-layer Chimaera network across its development. The network was presented with a sequential set of seven three-dimensional input vectors representing colours, three of which (green, blue, and red) were selected to be orthogonal, while the remaining four were pseudorandomly generated. As the SOM is essentially used as a high-dimensional filter capable of producing a low-dimensional activation map, from a computational perspective there is little difference in using low-dimensional or high-dimensional input vectors as “conceptual input” other than the ease of interpretation and speed of computation. As such, for demonstrative purposes these low-dimensional input vectors representing RGB (red, green, blue) colour triplets were used, which have a straightforward and intuitive interpretation as coloured regions on the data-vector maps. A summary of the parameters of this simulation is included in Appendix A.

<Insert Figure 2.1 about here>

Both the SOM learning algorithm as well as the Hebbian associative network are updated simultaneously at each epoch in the network – that is, the simulation does not contain separate stages for "SOM learning" and "Hebbian learning", with the caveat that the radius-of-effect σ of the SOM learning algorithm reaches zero at the 20th series of epochs of the network, and as such the SOM data vector representations will become stable and do not significantly change after this time. Because of this concurrent learning, the Chimaera network simultaneously acquires data vector representations for each node, as well as implicit temporal representations corresponding to sequence information in the associative network. Pseudo-code describing the Chimaera network algorithm is included in Appendix C.

2.3.3 Development of predictions

The development of a Chimaera network from initialization is depicted in Figure 2.1. The SOM learning rule's radius-of-effect σ is initially set to the size of the network (20 nodes, in this case), and is decreased by one at the beginning of each series of epochs. While the network's data vectors were initially set to pseudorandom values (see Figure 2.1, top left data vector map), the SOM's very large initial radius-of-effect had dramatically reshaped the data vectors on a network-wide scale by the end of the first series of epochs. While the data vectors are still nebulous at this initial series of epochs, by the 10th epoch series we begin to see separable regions form. By the 15th epoch series, the SOM radius-of-effect has decreased from 20 nodes to only 5 nodes, while having progressively altering the data vectors in local regions of the network to more closely resemble those in the input vectors. At this 15th epoch series, two critical transitions begin to appear in the network topology. First, discrete regions corresponding to individual input vectors in the sequence are now clearly visible in the data vector map. These regions also now resemble their corresponding input vectors to such a degree that the activation map, relatively silent until now, begins to come alive with *discrete* regions of activation. This signifies that the data vectors in the various regions of the Chimaera are now close enough to their respective input vectors that (10) produces significant levels of activation in a region when the input vector corresponding to that region is presented as input.

Decaying activation, while somewhat visible in the 15th series of epochs, begins to clearly emerge in the 20th series. This is, in part, an artifact – decay is always present in the network's dynamics, but the activation produced before the 20th series of epochs is not often above the value of the k_d parameter of (13), and as such, decays in only a single epoch. By the 20th series of epochs, the SOM radius-of-effect has decreased to zero, signaling that SOM learning is complete, and that the data vectors of the Chimaera network now very closely resemble those of the input vectors. As a result, (10) now produces very high levels of activation for regions of nodes when their corresponding input vector is presented. This high level of activation is now well above the value of the k_d parameter – so much more, that the activation takes more than one epoch to decay, and is clearly visible when the next input vector in the sequence is presented.

At the 25th series of epochs, we begin to see the effects of the Hebbian association map. While the association mechanism has always been active, only recently have its two preconditions been widely present in the network – namely, an activation map with two regions: one currently active region, and one previously active, decaying region. Following (14), the decaying region, having been previously active and above the k_{s_thresh} threshold on the previous epoch, will now preferentially associate with other active regions on the current epoch, which includes the currently active region. Over time, the association strength that builds between previous and current regions causes the activation from the *current* input vector (the next epoch's *previous* input vector) to flow to regions representing the *next* input vector (the next epoch's *current* vector). By the 30th series of epochs, these association weights have continued to increase, with more predictive activation visible. At this final series of epochs we can now clearly begin to see the full temporal dynamics of the network, where, for example, the activation map corresponding to the second input vector (blue) also contains decaying activation from the previous input vector (green), as well as activation predicting the next input vector (red) in the sequence. The implicit "*(t-1)* decay → *(t)* activation → *(t+1)* prediction" cycle in the activation map is illustrated in Figure 2.1 (inset).

< Insert Figure 2.2 about here>

The visualization shown in Figure 2.2 depicts the association map at the 20th series of epochs, well before the association weights have become sufficient to allow easily visible levels of activation to flow from one region to another. The global 20x20 array of Figure 2.2 depicts the individual association maps for each of the 400 nodes in the simulation. Examining the region of nodes representing the first input vector in the sequence (green, center), a cluster of approximately 8 nodes have easily visible association maps with two central regions. The smaller of these regions (center) corresponds to the location of the nodes representing the first input vector (green) itself, while the larger region (left center) of these individual association maps corresponds to the next vector in the sequence – namely, the second vector (blue). This signifies that the association map has associations from the region representing the *first* input vector to the region representing the *second* vector, as well as self-associations from the first input vector back to itself. This latter association occurs as a result of the general association mechanism in (14) being non-discriminatory about which nodes it chooses to associate with – if a given region was highly active on the previous epoch, it will associate with *any* regions of significant activation on the current epoch, including its own decaying activation.

Using both the local association maps of Figure 2.2, as well as the highlighted predictions of Figure 2.1 (inset) as a guide, one can follow the global association map of Figure 2.1 to see the exact sequence. As we have seen, the first input vector (green, center) associates to itself and the second input vector (blue, left center). Examining the second input vector's cluster of approximately 8 nodes with easily visible activation maps, we see this vector associates both with itself, as well as the region representing the third vector in the sequence (red, lower center). Following this pattern, the third vector (red, lower center) associates with the fourth (orange, bottom center), which associates with the fifth (purple, right bottom), which itself associates with the sixth (dark purple, right top). The sixth input vector associates with the seventh (dark blue, left top), which is the final input vector in the sequence, and as such, contains no further temporal associations.

2.4 Resolving Ambiguous Predictions with Multiple Layers

While the above simulation successfully represented a single simple sequence, the prediction mechanism does not generalize well to cases where the same data vector occurs more than once – either in the case of a single occurrence within multiple sequences, or multiple occurrences in a single sequence. In these cases, when at a repeated input vector, the prediction mechanism will generate multiple ambiguous predictions for the next node in the sequence corresponding to each of the possible next steps from that input vector across *all* sequences. In essence, the simple prediction mechanism is unable to use context – for instance, the previous elements in the sequence – to help localize the specific sequence being presented and generate an unambiguous prediction. In this section we examine how adding network layers can resolve the ambiguity in certain classes of sequences, making use of context information that is already present in the Chimaera's activation map.

2.4.1 The Processing Context

At each epoch the Chimaera network calculates a full activation map based not only on the classification of a best-matching node, but also taking the flow of activation through Hebbian association, and decaying transient activation into account. As such a *subset* of the information in this activation map includes information akin to the best-matching node for a given input vector, but the activation map is also rich in other information that includes the network's current "*processing context*" – namely, what has recently happened, and what is likely to happen next. It then seems natural to use this activation map itself as input to subsequent layers.

What exactly comprises the "processing context"? Let us recall that at a given epoch, the Chimaera architecture contains: (1) decaying activation corresponding to the previous input vector, (2) current activation corresponding to the current input vector, and (3) predictive activation corresponding to what input vector the network has previously transitioned to from the current input vector. For a given input vector, the current and predictive activations will be approximately identical – for example, if the network had been exposed to the sequences "blue-orange-green" and "purple-orange-red", when presented with the input vector corresponding to "orange", the regions corresponding to the "orange" data vector will activate, and will generate predictive activation for both the "green" and "red" regions. This will occur when the "orange" input vector is presented in both sequences, regardless of the prior input vector. However, as a result of the previous input vector, the network will also contain transient activation in the regions corresponding to either "blue" or "purple", respectively. As such, while the prediction for the next transition may be ambiguous – the network has activated both the "green" and "red" regions – the processing context (and activation map) are unique for a given transition *from* one input vector *to* another input vector, regardless of whether the subset of nodes that includes the prediction is unique. Described another way, we could say that each processing step corresponding to a transition from the *(t-1)*th input vector to the *t*th input vector will generate a unique activation map, regardless of the prediction for the *(t+1)*th input vector. Using this activation map as input to subsequent layers, we can essentially trace the path of the sequence back one step, such that in a multi-layer system, a higher-layer network will contain a *single vector* that represents the transition from the *(t-1)*th to *t*th nodes, rather than just a single *state*. This distinction is critical – where a first-layer Chimaera network's data vectors correspond to *states* such as "orange", data vectors in subsequent layers correspond to *transitions* in the previous layer such as "purple-orange".

The predictions and decays for subsequent layers will similarly be second order – while the first layer Chimaera will equally predict "green" and "red" as possible transitions when "orange" is the current input vector, the second-layer will correctly and unambiguously predict the processing context associated with an "orange-green" transition as the next transition in the sequence when "blue-orange" preceded it, while "orange-red" will similarly be predicted when "purple-orange" had first been presented.

<Insert Figure 2.3 about here>

An interesting possibility soon emerges when we consider the general case of a multilayer network with an arbitrary number of layers. A one-layer network is capable of ambiguously predicting all possible transitions from a given state, but its processing context is capable of representing the previous processing state – essentially a second-order feature. As such, a second-layer network using this first-layer processing context as input is capable of unambiguously predicting 1-back sequences – sequences whose next transition depends exclusively on the previous transition. However, as in the case of the first-layer network, this second layer network's processing context contains two pieces of information: the current transition, and the previous transition. As such, a third-layer network would be capable of unambiguously predicting 2-back sequences – sequences whose next transition can depend upon up to two of the previous transitions (such a sequence is shown in Figure 2.3). In the general case, we might posit that an n^{th} layer Chimaera network is capable of unambiguously representing and predicting an *(n-1)*-back sequence, while its processing context contains the information required to construct an *n*-back path.

2.4.2 Multilayer Simulations 2.2 and 2.3

Figures 2.4 and 2.5 depict two multilayer Chimaera network simulations, one two-layer, and one three-layer, trained on the sequences described in Figure 2.3. The two-layer, 1-back Chimaera network was presented with a series of two sequences (see Figure 2.3a) that together formed the 1-back ambiguity centered around the "orange" vector, described in the previous section. Figure 2.4, series 60 highlights the predictions for each input vector. Note that when the orange input vector is presented to this network, the first layer is unable to resolve the transitional ambiguity and generates multiple predictions for the next transition. The second layer of this network, whose input comprises the processing context of the first layer, contains separate representations for 1-back transitions instead of single states. In the case of sequence 1, these 1-back transitions consist of "start-blue", "blue-orange", and "orange-green", respectively, while in the case of sequence 2, the 1-back transitions consist of "start-purple", "purple-orange", and "orange-red". As a result of representing these 1-back transitions instead of single states, the second layer is able to both represent the possible transitions to the ambiguous "orange" vector, as well as unambiguously predict a single transition from the "orange" vector based on prior experience.

<Insert Figures 2.4 and 2.5 about here>

Similarly, Figure 2.5 depicts a three layer Chimaera network whose input comprises the two 2-back and one 1-back sequences in Figure 2.3b. In this case, a mechanism of 1-back lookback is unable to resolve the ambiguity when the previous transition was "purple-orange" – one must look one transition further, to "yellow-purple-orange" or "cyan-purple-orange", to determine whether the next transition will be to "red" or "green". As such, the first two layers, corresponding to 0-back and 1-back sequences respectively, generate ambiguous transition predictions, while the third layer of the Chimaera network generates unambiguous predictions for this set of 2-back sequences. A summary of the parameters used in these simulations is included in Appendix B.

2.5 Limitations and Discussion

Several factors are critical to maintaining the dynamics of the Chimaera architecture. In addition, as a test framework, several other issues are present with the current architecture that limit its representational capabilities and general applicability as an alternative SOM architecture for storing arbitrary sequences. A number of these issues are described below.

2.5.1 Representing Data Vectors

Kohonen's (1982) self-organizing map learning algorithm uses a radius of learning σ that progressively decreases over successive epochs, resulting in data vectors that both very closely represent input vectors, as well as a separable and discrete topology of data vector regions where similar data vectors are represented topologically close. Unfortunately this decreasing radius results in a network that, after a time, is unable to represent new input vectors. Ideally an experiental and developmentally-plausible architecture should be able to acquire new representations whenever they might be available. A modified learning method where the radius-of-learning depends upon a given input vectors relative representation in areas nearby the best-matching node – similar to the calculation of a similarity map – may be one potential method to designing adaptive learning rules that continually allow new representations to be acquired while adjusting the radius of learning so as not to completely overwrite previously stored representations.

2.5.2 Topography can affect activation

The method of Hebbian-inspired association used in this architecture associates nodes that were significantly activated on the *previous* epoch with nodes that are significantly activated on the *current* epoch. In subsequent epochs, activation will then directionally flow from the former to the latter region, proportional to the association weight between these two regions. In cases where a large disparity exists between the *number* of nodes representing the former and latter data vectors, two issues may occur: (1) activation from the region with few nodes will flow into the region with many nodes, potentially causing less activation in "prediction" regions than would otherwise occur, or (2) activation from the region with many nodes will flow into the region with few nodes, potentially causing hyper-activation to occur that, in the worst case, may persist over many trials. This latter problem is of particular concern as, over time and in some cases, associations can build to and from these hyper-activated nodes, eventually causing activation to pool in an area, and the temporal dynamics of the network to no longer function as desired. Fortunately this issue is only likely to occur in known circumstances where a given input vector occurs significantly more or less than the average presentation frequency of all vectors. However, artificially constraining a dataset to approximately balanced presentation frequencies does introduce a developmental implausibility to the model, and suggests the learning rule for the network is as of yet incomplete. The suggested solution to the problem of *representing data vectors* in self-organizing maps (outlined above) would likely result in a network where each data vector received approximately equal numbers of nodes (nearly irrespective of presentation frequency), mediating or solving this issue entirely.

2.5.3 Autonomous sequence "play-back"

Currently temporal sequence learning in the Chimaera architecture takes the form of a prediction for the next item in the sequence. Given a single input vector at the start of a sequence, the network is currently unable to "play back" that sequence by way of progressively activating the nodes representing successive data vectors in that sequence. This is due, in large part, to the relatively low level of activation that transfers from a highly active node to its associated nodes.

In the future, refinements to the temporal and associative dynamics of the network may allow this predictive activation to increase – ideally to nominal levels – allowing autonomous sequence playback to take place. These refinements may be relatively simple and take the form of inhibiting self-association, which is a likely sink for activation flow between nodes. The methods may also be more computationally interesting, and one might imagine additions to the prediction mechanism that take the form of activation feeding back down from higher layers in a multilayer Chimaera network. This latter method has the additional benefit of potentially assisting with ambiguity resolution at lower-levels, based on higher-order unambiguous predictions.

2.5.4 The relation between activation level and ambiguity resolution

The ambiguity resolution mechanism of multilayer Chimaera networks critically depends on the activation map that serves as input to an n^{th} layer to be unique in as much as the network is capable of differentiating between different activation maps, where this capability is defined as follows.

At an ambiguous transition in layer *(n-1)*, several factors determine what the input vector for layer *n* will look like, including: (A) the transient, decaying activation from the previous input vector to layer *(n-1)*, (B) the current activation from the current input vector to layer *(n-1)*, and (C) the activation corresponding to the $(n-1)^{th}$ layer's predictions for the next transition. Of these three considerations, (B) and (C) will be approximately equal for all ambiguous transitions from a given node, while (B), then (A), respectively, have the highest activations, and hence the greatest impact on determining the *difference* or *distance* between two or more ambiguous input vectors to the n^{th} layer. As (B) is relatively constant across a given set of ambiguous transitions, the largest contributor to ambiguity resolution in the n^{th} layer, and the ability for the n^{th} layer to discriminate between ambiguous input vectors, is the transient decay of activation in the $(n-1)^{th}$ layer. As such, if the parameters of the Chimaera network are ever such that this transient, decaying activation is insufficient for (10) to generate non-zero values for *only* one input vector in a given set of ambiguous transitions, then the n^{th} layer will be unable to resolve the ambiguity of those vectors.

2.6 Summary and Contexts

The Chimaera framework has successfully tested the alternate dual-representational scheme of representing conceptual information and sequence information in separate forms, with separate processing mechanisms. While conceptual knowledge is represented in the form of data vectors containing feature information, temporal associations between data vectors are represented with a progressive association mechanism that strengthens over time with repeated presentation. This simple association mechanism generates ambiguous predictions in cases where a given data vector occurs more than once across all sequences. For sets of sequences that can be adequately represented as *n*-back sequences, where *n* is the number of layers in a multilayer Chimaera network, the activation maps of subordinate layers allow higher layers to represent "transitions between concepts", and resolve this ambiguity. For complex sequence sets where the number of layers becomes large, this style of representation and processing may be impractical.

In the context of previous temporal representation schemes, this framework essentially uses leaky-integrators to create tapped-delay lines. This is possible by implementing a decaying activation map at the output of the SOM, which then serves as input to superordinate layers in a multi-layer network. Where the activation map is analogous to a large array of leaky integrators, subsequently using this information as input to superordinate layers generates a method of *n-1* lookback, the essence of iteratively constructing a tapped-delay line. As such, an *n*-layer Chimaera network contains the mechanisms of an *n-1* ambiguity resolution mechanism, or a tapped-delay line with a window size of *n-1*. This bears aspects of temporal sequence representation used across the TKM architecture, SARDNET, as well as RecSOM.

While the Chimaera SOM is currently only a test framework for the idea of a self-organizing dual-representational system, and has a significant way to go before it is generally-usable as an architecture for temporal representation, the ideas that it tests continue to be of interest for modelers. The success of Chang et al. (2006)'s dual-representational model of morphosyntactic use underscores that in addition to architectures serving as a representational substrate, in terms of cognitive modeling it is also important that representations of a specific modality are able to interact with and be used by cognitive processes. In terms of processes, it would be particularly interesting to apply unsupervised architectures such as the Chimaera to generating models of morphosyntactic use, similar to the supervised model of Chang et al. While there is often a motivation, in part due to the self-organizing map's intrinsic developmental properties, to model the acquisition and development of representations over time (e.g. lexical development, Li, Farkas, and MacWhinney, 2004), or to model the process of classification (e.g. phoneme classification, Kohonen, 1988), it would be particularly interesting to use the Chimaera SOM or a similar unsupervised dual-representational framework to explore more complex processes of a non-classification nature, such as the generative processes of psycholinguistic production. While some aspects of these processes, such as the selection of a phonetic representation for a given concept, may be reduced to classification mechanisms, modeling human performance during the online generation of a sentence-level grammatical structure *and* the fusion of that structure with the contents of a mental model, similar to Chang et al., is subject to a variety of non-intuitive mechanisms such as structural priming (Bock and Griffin, 2000) that are likely unable to be captured with current mono-representational temporal SOM architectures. Similarly, the associative mechanism used by the Chimaera to queue temporal sequences – a process that generates ambiguous context-independent predictions – while not being particularly useful for generating unambiguous predictions of *specific* sequences, bears a close functional similarity to the idea of priming. As such, dual-representational SOM models making use of the combination of an explicit vector representation mechanism with an implicit associative mechanism may allow complex models of temporal sequence processes, while paving the way for incorporating non-classification processes such as priming into unsupervised models of cognitive performance.

Appendix A

The single-layer Chimaera simulation of Simulation 2.1 consisted of the following parameters: 2D spatial array, 20x20, torus topology, 3 data dimensions initialized to pseudorandom values; activation decay between epochs (k_d): 0.50; total activation for a given node clipped to a maximum of: 2.5; k_m: 0.010; k_{s_thresh} threshold = 0.8; noise threshold: 0.1; distance scale (τ): 0.10; Initial SOM learning radius-of-effect: 20 (network size), decreased by 1 at the beginning of each series of epochs. SOM Similarity map: radius of 3. Neighborhood function h_{ij} took the form of a linearly-tapered window.

Appendix B

The multilayer Chimaera simulations (Simulations 2.2 and 2.3) consisted of the following parameters: 2D spatial array, 20x20 (2-layer), 30x30 (3-layer), both torus topologies, all layers initialized to pseudorandom values; In both simulations: 3 data dimensions (layer 1), 400 data dimensions or 900 data dimensions in subsequent layers for 2-layer and 3-layer network, respectively; activation decay between epochs (k_d): 0.25; total activation for a given node clipped to a maximum of: 2.5; k_m: 0.010; $k_{s\_threshy}$ threshold = 0.8; noise threshold: 0.1; distance scale (τ): 0.10; Initial SOM learning radius-of-effect: 20 or 30 (network size), decreased by 1 at the beginning of each series of epochs, where subsequent layers began their SOM training after their input layer ended training. SOM Similarity map: radius of 3. Neighborhood function h_{ij} took the form of a linearly-tapered window.

Appendix C

```
for epoch = 1 to network_size
     for current_input_vector = 1 to sequence_length

          Select next input vector in sequence

          SOM Learning rule for data vectors {
               Find best-matching data vector
               Modify all other nearby data vectors to more
                    closely resemble input vector
          }

          Compute Activation Map {
               Compute activation based on each data vector's
                    similarity to the input vector
               Add previous activation map (minus a decay
                    factor)
               Allow activation to flow to other nodes based on
                    the Hebbian association map
               Limit activation values (e.g. clip to between
                    zero and one)
          }

          Modify Hebbian Association Map {
               Compute change in associativity from each node,
                    to each node.  Only build associations from
                    nodes that recently fired, to nodes that are
                    above the noise threshold.
          }

          Decrease SOM learning rule's radius-of-effect

     end for
end for
```

Figure Captions

Figure 2.1: The single-layer Chimaera network of Simulation 2.1, shown at regular intervals from epoch series 1 through 30. The initial random state of the network is shown at the top of the data vector column. Similarity maps: Dark regions refer to areas of similarity, while light boundaries describe transitions between regions. (Inset) The implicit process of (n-1) decay, (n)

current activation, and (n+1) prediction is shown overlayed upon an enlarged set of activation maps from epoch series 30.

Figure 2.2: A four dimensional association map for the single-layer Chimaera network depicted in Figure 1, at epoch series 20. This activation map consists of a two dimensional (global) array of two dimensional (local) arrays, representing the association from a given node (global location) to a given node (local location). Each set of nodes corresponding to some similar data vector tend to be associated both to their own region, as well as to the region corresponding to the next set of nodes in the sequence. Lighter regions depict proportionally more association weight, where the entire map is scaled such that the brightest value reflects the highest connection weight across the entire map.

Figure 2.3a: A set of two sequences with a single, common node (orange). Correctly predicting the next transition in the sequence from orange requires looking back one transition.

Figure 2.3b: A set of three sequences, two of which with two common nodes (purple-orange), and all three with a single common node (orange). Correctly predicting the next transition in the sequence from orange requires looking back two transitions, in the case of sequences two and three.

Figure 2.4: The two-layer Chimaera network of Simulation 2.2, shown at regular intervals from epoch series 1 through 60. The two input sequences to this network correspond to the 1-back sequences depicted in Figure 2.3a. At series 60, Layer 1 is unable to resolve the 1-back transitional ambiguity, and generates two predictions. Layer 2 is able to resolve the 1-back ambiguity, and generates only a single prediction.

Figure 2.5: The three-layer Chimaera network of Simulation 2.3, shown after training at epoch series 128. The three input sequences to this network correspond to the 2-back sequences depicted in Figure 2.3b. At epoch series 128, Layers 1 and 2 are unable to resolve the 2-back transitional ambiguity, and generate two predictions. Layer 3 is able to resolve the 2-back ambiguity, and generates only a single prediction.

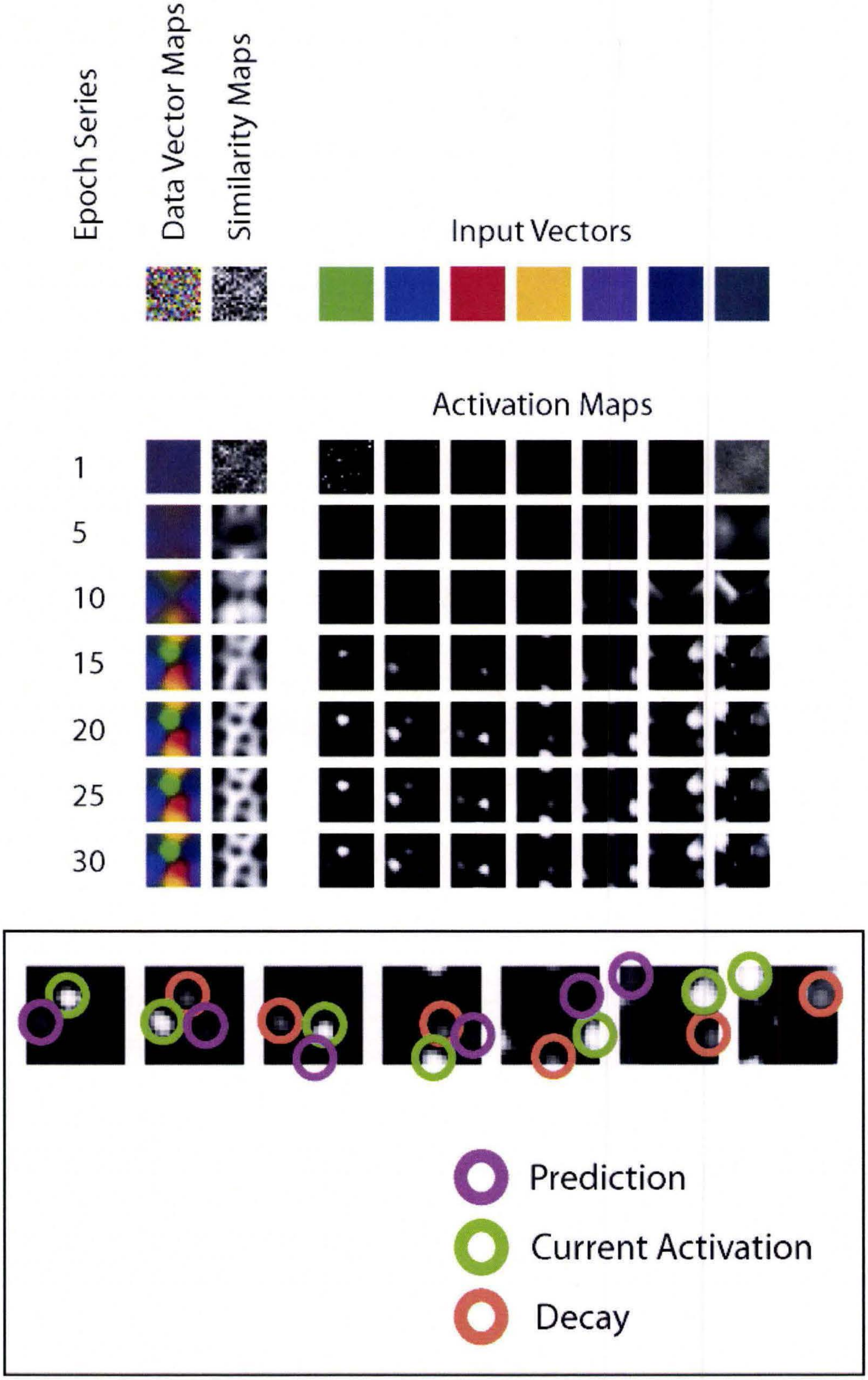

Epoch Series
Data Vector Maps
Similarity Maps
Input Vectors
Activation Maps
1
5
10
15
20
25
30
Prediction
Current Activation
Decay

Figure 2.1

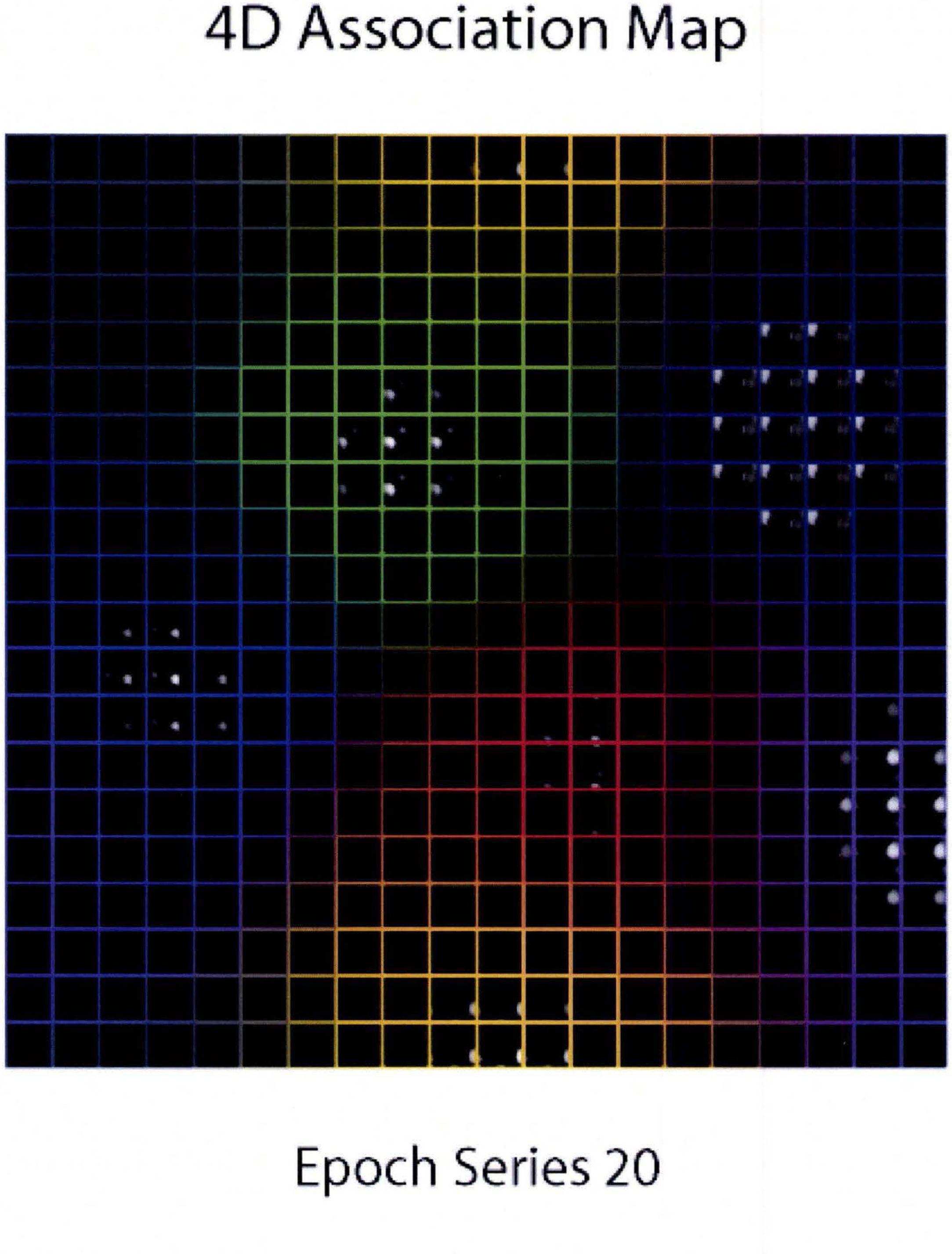


Figure 2.2

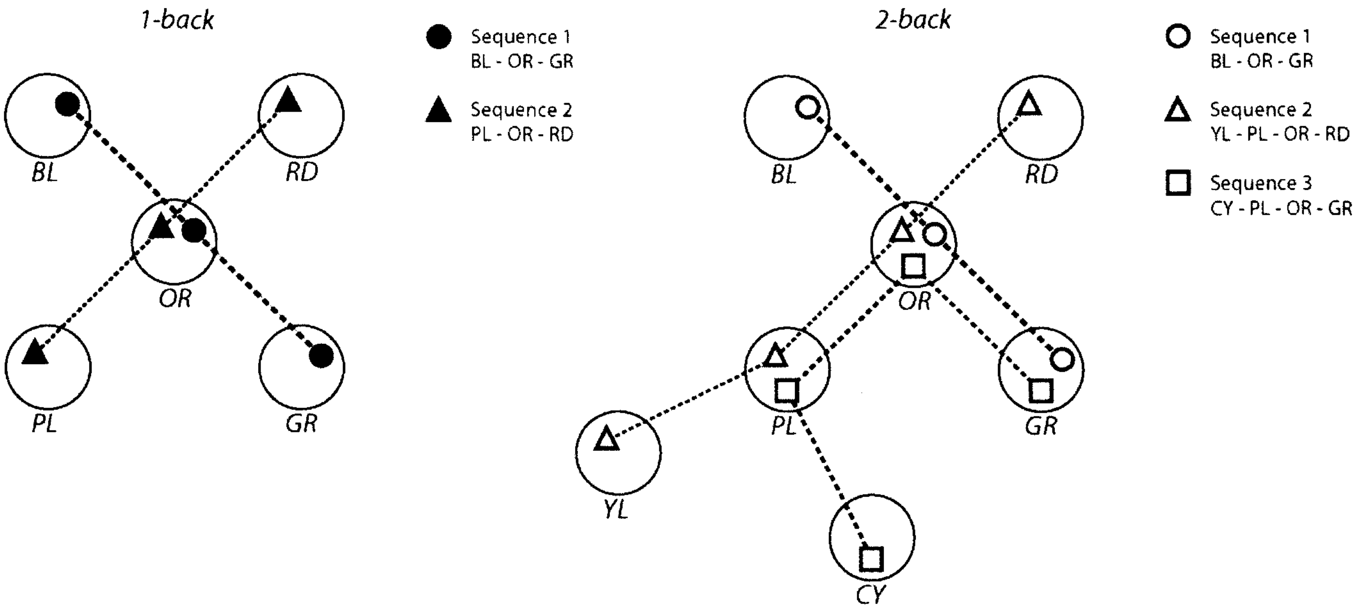

1-back
● Sequence 1
BL - OR - GR
▲ Sequence 2
PL - OR - RD
BL
RD
OR
PL
GR
2-back
○ Sequence 1
BL - OR - GR
△ Sequence 2
YL - PL - OR - RD
□ Sequence 3
CY - PL - OR - GR
BL
RD
OR
PL
GR
YL
CY

Figure 2.3

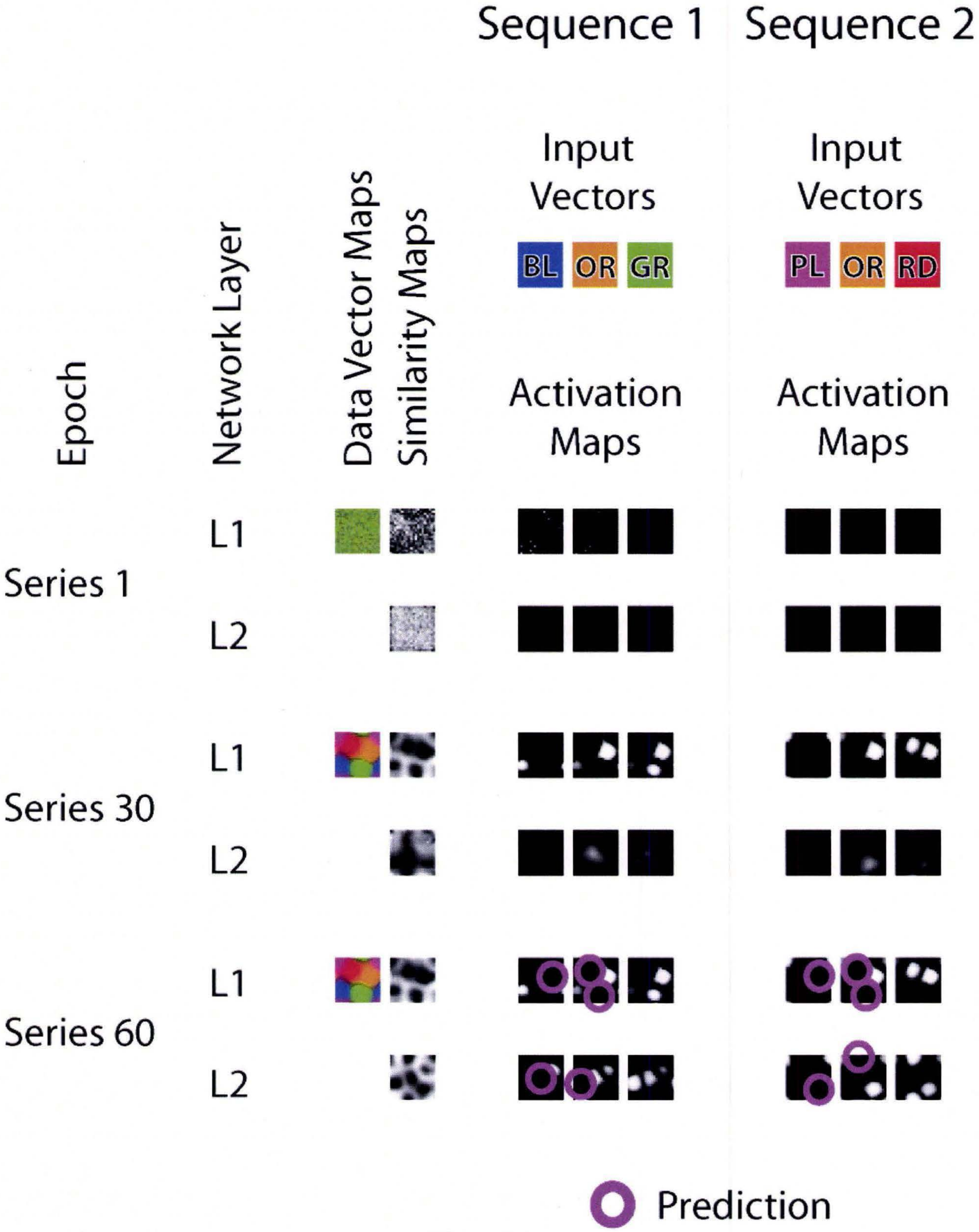

Sequence 1
Sequence 2
Epoch
Network Layer
Data Vector Maps
Similarity Maps
Input Vectors
BL
OR
GR
Input Vectors
PL
OR
RD
Activation Maps
Activation Maps
Series 1
L1
L2
Series 30
L1
L2
Series 60
L1
L2
Prediction

Figure 2.4

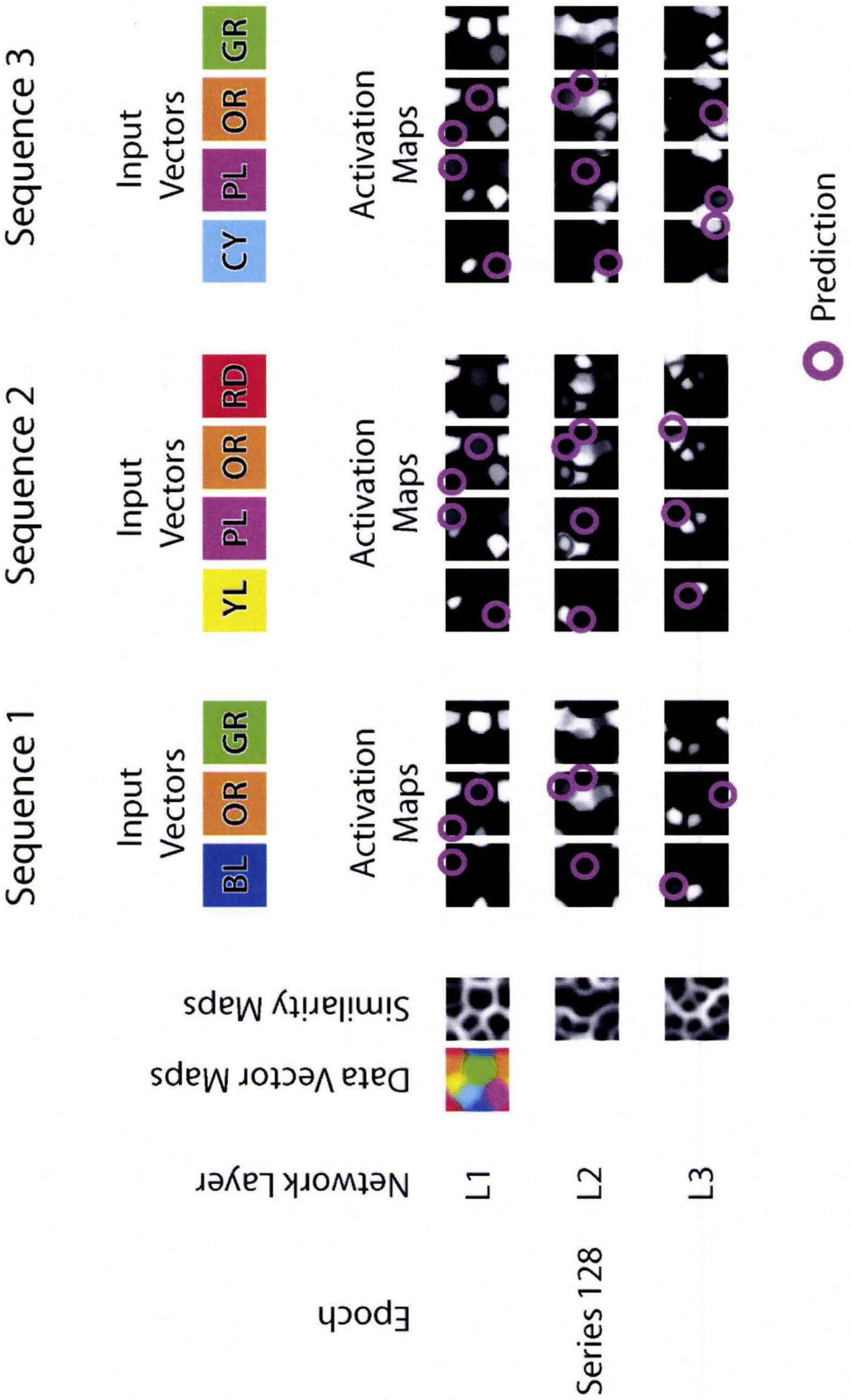

Epoch
Series 128
Network Layer
L1
L2
L3
Data Vector Maps
Similarity Maps
Sequence 1
Input Vectors
BL
OR
GR
Activation Maps
Sequence 2
Input Vectors
YL
PL
OR
RD
Activation Maps
Sequence 3
Input Vectors
CY
PL
OR
GR
Activation Maps
Prediction

Figure 2.5

Chapter 3

Multi-layer Chimaera Networks for Self-Organizing Linguistic Grammar Acquisition

3.1 Introduction

Contemporary connectionist modeling of grammatical acquisition and linguistic processes that use grammatical knowledge have had numerous success stories, and have developed interesting models of grammar acquisition and use. Because of the difficulty of this task, our models have tended to use supervised neural network architectures with very capable learning rules, such as the Simple Recurrent Network (SRN; Elman, 1990), while our datasets generally have limited grammatical complexity – often being constrained to only two to four parts of speech, with few unique sentence structures. This project investigates the use of self-organizing neural networks for linguistic grammar acquisition, examining both to what extent an unsupervised architecture based on correlational learning (rather than supervised error-based learning) can be used to acquire grammatical knowledge, as well to investigate the self-organizing high-order representational performance of the network on a somewhat more complex grammar training set that includes eight parts of speech and 20 sentence structures.

The simple recurrent network (Elman, 1990) is a supervised neural network architecture that incorporates temporal extensions to the 3-layer feed-forward architecture of Rumelhart et al. (1986). As Elman (1990) noted, these temporal extensions make the network particularly suitable for processing temporal sequence information, and in particular, linguistic streams. After Elman's initial success, the simple recurrent architecture has since been successfully applied extremely broadly to a variety of language-learning tasks, including abstract (e.g. Frank et al., 2006) and sensorimotor-grounded (e.g. Howell et al., 2005) grammar learning, and, with extensions, structural priming based on the fusion of abstract syntactic knowledge with semantic representations (e.g. Chang, Dell, and Bock, 2006), to name only a few of the many success stories.

Simple recurrent networks, powered by the backpropagation of error algorithm (Rumelhart et al., 1986), are particularly good at learning sequences, associations between arbitrary input vectors, and grammars that can be expressed in the form of a finite state automation (see Chapter 4). Computationally, the links and associated connection weights between the layers of an SRN represent a transduction process, whereby a pattern of activation presented to the network at its input layer – representing some arbitrary input pattern, such as a phoneme, morpheme, or word, for example – will progressively flow through the network, producing a corresponding output pattern on the nodes of the output layer. This output will correspond not only to the current input of the network, but also to the temporal context of that input including vectors that were presented earlier in the linguistic stream. Representations of a given aspect of knowledge, whether these be at the level of phonemes, words, or concepts, come about as transient patterns of activation on the network's nodes – and in this sense, the network has explicit representations of a transduction process that will bring about one sequence element given another element in context, but it has no explicit (in terms of being implemented in connection weight) representations of the concepts or words it represents sequences of – it contains only information on how to proceed from one word to another, given a certain temporal location in a sentence.

A self-organizing map (SOM), in comparison, is an unsupervised neural network architecture developed by Kohonen (1982) as a computational analog to the topographic organization observed in perceptual cortex, where representations of similar percepts (for instance, tones of similar frequency, or lines of similar orientation) are often observed in topographically close areas of cortex. In comparison to a simple recurrent network, a SOM can be thought of as containing the other half of a complete computational/representational system – it is particularly good at explicitly representing arbitrary perceptual, conceptual, or abstract knowledge specified mathematically as a vector, but without extensions is virtually unable to carry out processing with those representations.

Many extensions to the self-organizing map exist, particularly those that add temporal sequence processing through recurrent mechanisms (eg. TKM, Chappell and Taylor, 1993; RecSOM, Voegtlin, 2002; SOMSD, Hagenbuchner et al., 2003), similar to Elman's addition to the 3-layer feed forward architecture. Here, we investigate the use of one of those extensions, the Chimaera network (Chapter 2), which is capable of explicitly representing both sequence elements as well as the temporal associations between those elements, for the unsupervised correlational learning of a linguistic grammar data set.

3.2 Methods

3.2.1 Input Set

In order to examine the Chimaera's capacity for grammar learning across a variety of different linguistic sentence structures, an input set of sentence parses specified at the part-of-speech level was constructed. These 20 full sentence parses contained grammatical (English) combinations of up to 8 parts of speech (including DETERMINER, ADJECTIVE, NOUN, VERB, PREPOSITION, ADVERB, AUXILIARY, and CONJUNCTION), and ranged in length from 2 to 8 parts-of-speech. This linguistic input set is itself expressed at a high-level, representing general grammatical combinations of parts of speech rather than specifying specific instances of grammatical sentences at the lexical level. The 20 sentence parses principally consisted of several examples of noun phrases, verb phrases, and preposition phrases, instantiated in various combinations. As such, often times a given sentence parse may be contained, wholly or in part, in another sentence parse, as in the case of the parses (DET – N – V, as in the sentence "the cat ate", and DET – N – V – DET – N, as in the sentence "a girl picked an apple"). The depth of ambiguities in this first linguistic grammar training set were such that each grammatical transition could be correctly predicted using a bigram model, such that deeply ambiguous sequences were not present. The input set is described in Table 3.1.

For ease of visualization, each of the eight parts of speech was mapped to a corresponding vector containing three dimensions, one each for red, green, and blue. Care was taken to ensure the values of these vectors were chosen to be relatively distant from each other, such that the activation function of the network would not classify any of the 8 part-of-speech vectors as similar to one another – a situation that would cause interference when acquiring temporal information.

3.2.2 Architecture

The simulation used the 3-layer Chimaera network architecture described in Simulation 2.2 . Following the training behaviour identified in Chapter 2, the first layer of this network was to acquire representations of both the individual parts-of-speech present in the sample grammar, as well as transitions between these parts of speech. To facilitate higher-order representational clustering, second and third layers were included, where the training of a given layer began after training in the previous layer had stabilized. This architecture is described in Figure 3.1.

Further, as the Discussion of Chapter 2 identified several issues with the association mechanics of the Chimaera contributing to relatively low levels of associative flow, the association mechanics of the architecture were further refined to prevent the self-association of a given data vector with either itself or nearby neighbors that represent the same data vector. The function governing the association mechanics was modified to prevent association in cases where the two nodes associating contained very similar data vectors (evaluated as a difference in the Euclidian distance between the two vectors):

$$\begin{aligned} &\Delta a_{ji}(t) = k_m \cdot k_s(j,t) \cdot a_i(t) \cdot a_j(t), \\ &\quad for \left\| a_i(t) - a_j(t) \right\| > a_{thresh} \\ \\ &\Delta a_{ji}(t) = 0, \\ &\quad for \left\| a_i(t) - a_j(t) \right\| \le a_{thresh} \end{aligned} \tag{1}$$

where $\boldsymbol{a_{thresh}}$ represents the minimum Euclidian distance required between a given set of vectors before association can occur. This mechanism effectively preventing self-association, and allowed the magnitude of associations between temproral associates to increase.

3.3 Simulation 3.1

3.3.1 Layer One: Part of speech representations, as well as grammar representations

The results of the simulation are shown in Figure 3.2. The first layer of the network rapidly self-organizes from an initially random state, with separate representations easily visible for each of the 8 parts-of-speech included in the sample dataset visible and well-formed by epoch 20, when the radius of learning for layer 1 had decreased to zero, effectively stabilizing the representations contained within this layer. Here, for ease, the representations of each of the 8 parts of speech are colour coded, where (for example) the representation of "determiner" is coded as purple, "noun" as blue, "verb" as red, and so forth.

Figure 3.2 includes activation maps for one of the 20 input sentence structures, in this case the sentence parse "DETERMINER – NOUN – VERB – PREPOSITION – DETERMINER

– NOUN – ADVERB", potentially corresponding to a sentence such as "the cat ate in the house quietly". The activation for individual representations of parts of speech begins to emerge near epoch 10, is present for each part of speech by epoch 15, and is in its final form at epoch 20. By epoch 30, predictive associative activation from a given active node to nodes that represent possible grammatical transitions for the next element in the sequence are clearly present, although faint – this activation pattern is enlarged in Figure 3.3.

Hand tagging and analysis of the predictive activation showed the network had acquired the sample grammar provided in the 20 sample input sentences, where the activation maps in Figure 3.3 have been tagged to highlight the temporal process of activation, prediction, and decay across the grammar. Starting with the first determiner, the activation map contains both current activation highlighting the representation of determiner in layer 1, as well as predictive activation for the possible grammatical transitions from determiner to the next sequence element – in this case, either to adjective (as in the sentence "the soft…"), or to noun (as in the sentence, "the cat…"). Similarly, when the next element in the sequence is selected (noun), the network contains decaying activation for the element that it just transitioned from (determiner), current activation for the currently active sequence element (noun), and predictive activation for the next possible element in the sequence – here, one of verb, preposition, auxiliary, adverb, or conjunction. Similarly, for sequence element 3, verb predicts preposition, and preposition further predicts determiner, which then predicts noun, which finally predicts adverb, and completes the successful prediction of the sentence parse. Note that the magnitude of the prediction – represented in these visualizations by the intensity of the activation level – is a function of the presentation frequency of a given transition in the input set. As such, transitions that are relatively common in the input set (such as "DETERMINER – NOUN", or "NOUN –

VERB") will have higher associative weights, and correspondingly higher activations than transitions that are particularly infrequent in the data set (such as "NOUN – PREPOSITION", or "NOUN – AUXILIARY").

3.3.2 *Layer Two and Three: Representing transitions between parts of speech*

Multiple layers represent progressively higher-level features in the data – in this case, transitions between parts of speech. Figure 3.4 introduces a new visualization, a hybrid association map overlayed upon a similarity map, allowing both broad data vector clustering and structure, as well as temporal sequence information to be contained within a single visualization. This map has also been tagged to show the locations of data vectors representing transitions between parts of speech.

Layer 2 shows broad clustering based upon the current part-of-speech, where all the transitions to a given part-of-speech tend to cluster into topographically close areas. For example, all the possible transitions to a determiner (either from the beginning of a sentence parse, or from a verb, preposition, auxiliary, or conjunction) tend to cluster together in the top centre portion of layer 2, where similarly the possible transitions to a noun (either from the beginning of a sentence, or from a determiner, adjective, verb, or auxiliary) cluster in the bottom right portion of the network. By layer 3 (Figure 3.5), most of the iterative structures in the sample English grammar (such as the components of a noun phrase) are represented explicitly and clustered in topographically close areas of the SOM, while recursive rules (such as a preposition phrase) are represented implicitly in the association mechanics of the network.

3.4 Summary, Discussion, Future Directions

In summary, through exposure to examples, the unsupervised network acquires explicit representations of both the grammar, word-level part-of-speech categories, as well as many higher-level part-of-speech categories and rules that represent valid grammatical transitions within the sample data set.

For simplicity, the model currently accepts pre-parsed grammatical information at the part-of-speech level (e.g. DETERMINER – NOUN – VERB) rather than at the lexical level (e.g. THE CAT ATE), where the network would first have to "distill" part-of-speech information from a lexical-level sentence representation in order to arrive at a sentence parse to use as input. While the notion of discovering underlying structure in a diverse input stream is a separable problem from the idea of acquiring a temporal sequence, many contemporary supervised architectures tend to tackle both of these problems simultaneously, and at least some have achieved generally good performance with variations on Elman's recurrent network architecture (e.g. Frank, 2006), or other recurrent architectures (e.g. Farkas and Crocker, 2008). This is a substantial limitation of the current model, where distilling part of speech from word representations would significantly increase the model's utility, and make a much more plausible and general model of grammar acquisition from the linguistic stream, rather than from specialized pre-parsed linguistic input.

Further, while the data set used here includes far more diversity than is commonly present in grammar simulations in terms of both the number of parts of speech and the diversity and length of sentence structures, the underlying grammar does not contain any deep clauses or ambiguities, such that the next element in a given parse can be unambiguously predicted from the current position in that parse. For comparison, many models of grammatical systematicity in neural network processing make use of grammars that contain deep transitional ambiguities, such as the Van der Velde et al. (2004) grammar, which contains only three parts of speech (noun, verb, and "who" – a clause marker), as well as three short but deeply-ambiguous sentence structures including a simple sentence (N – V – N), as well as right-branching (N – V – N – Who – V – N), and centre-embedded (N – Who – N – V – V – N) clauses. While in principle a multi-layer Chimaera network with 4 layers, corresponding to the deepest grammatical ambiguity across these three structures ("<Who> – <N> – <V> – V" in the centre-embedded clause) would be able to acquire this simple grammar, it would take a Chimaera with many more layers to learn a grammar with similar clauses embedded in sentences with our significantly more combinatorally diverse 8 part-of-speech grammar – the number of combinations would require a mechanism of deep lookback, which would place a requirement for an implausibly large number of layers in a suitable Chimaera network. Clearly, as a successful model of even moderately ambiguous grammar learning, the Chimaera network structure must migrate from a feed-forward architecture, requiring an addition layer for each processing step in a looping process, and migrate to a much more compact, capable, and biologically-plausible recurrent architecture.

Sequence Element							
1	2	3	4	5	6	7	8
DET	N	V					
N	V						
ADJ	N	V					
DET	N	V	DET	N			
DET	N	PREP	DET	N	V		
DET	N	V	PREP	DET	N	ADV	
DET	N	CONJ	DET	N			
DET	ADJ	N	CONJ	DET	N	V	
V	CONJ	V					
DET	ADJ	N	V				
DET	ADJ	N	PREP	DET	ADJ	N	V
DET	N	PREP	DET	N	V		
N	PREP	N	V				
DET	N	AUX	V	N			
DET	N	V	DET	N			
DET	N	V	DET	ADJ	N	ADV	
AUX	DET	N	V				
AUX	N	V					
AUX	ADJ	N	V	ADV			

Table 3.1: The sentence structure parses used as input to Simulation 3.3. This set includes both full sentence parses, as well as parses that may serve as grammatical subsets of other parses.

Figure Captions

Figure 3.1: A schematic diagram of the 3-layer Chimaera architecture used in Simulation 3.1 . Layer 1 acquires representations at the part of speech level, where layers 2 and 3 acquire progressively higher-order representational features, representing transitions between parts of speech in the grammar.

Figure 3.2: Layer 1, shown at regular intervals from epoch series 1 through 30. The initial random state of the network is shown at the top of the data vector column. Similarity maps: Dark regions refer to areas of similarity, while light boundaries describe transitions between regions.

Figure 3.3: The implicit process of (n-1) decay, (n) current activation, and (n+1) prediction is shown overlayed upon an enlarged set of activation maps from epoch series 30.

Figure 3.4: A four dimensional association map for layer 2. Labels represent transitions between parts of speech, in the form "<FROM> TO". The clustering topography shows a preference for transitions to a given part of speech.

Figure 3.5: An association map for layer 3. Labels represent the overall category of representations clustered within a given spatial region. The clustering topography resembles phrase structure categories, where many of the components of (for example) a noun phrase tend to cluster together toward the bottom right of the network.

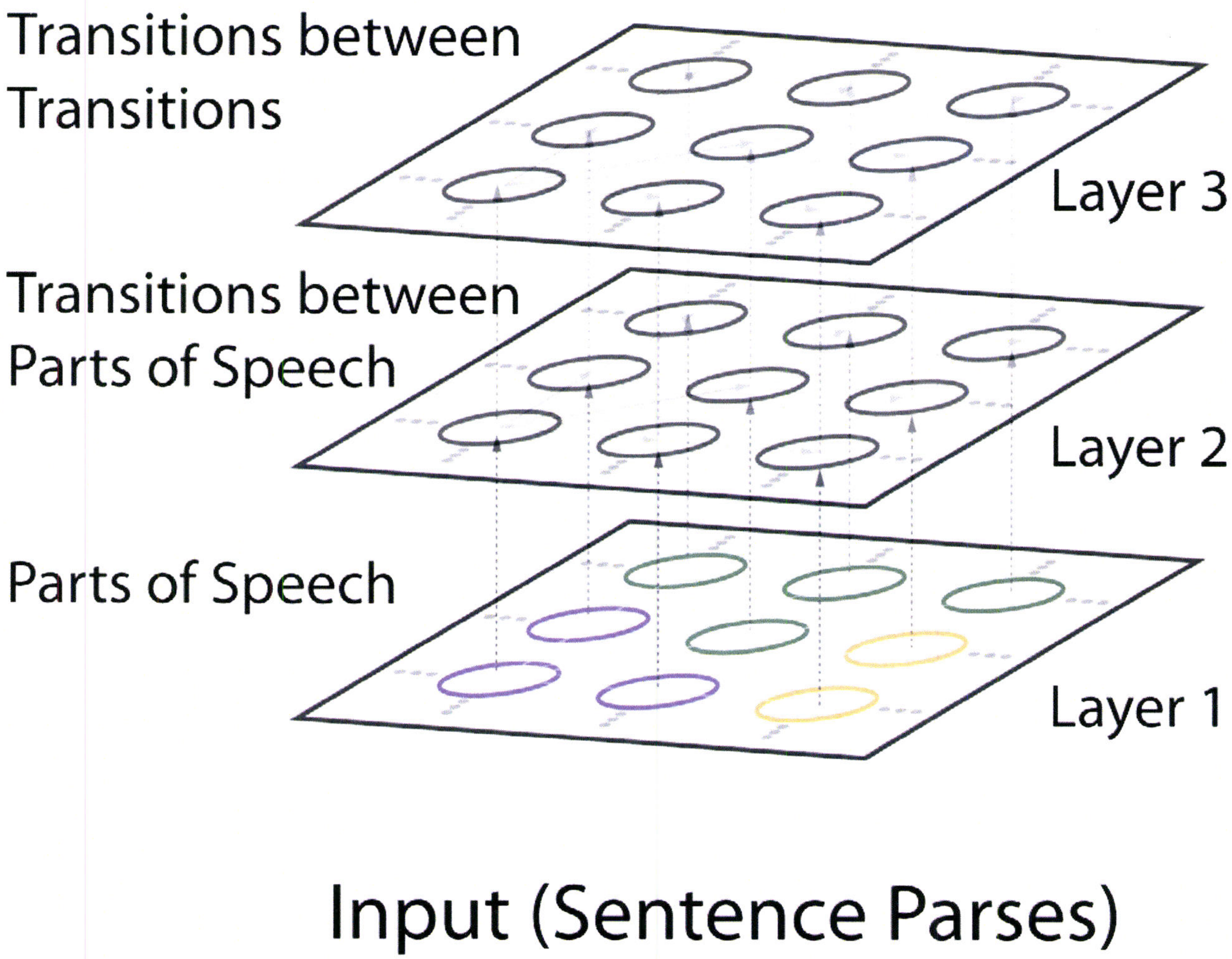

Transitions between Transitions
Layer 3
Transitions between Parts of Speech
Layer 2
Parts of Speech
Layer 1
Input (Sentence Parses)

Figure 3.1

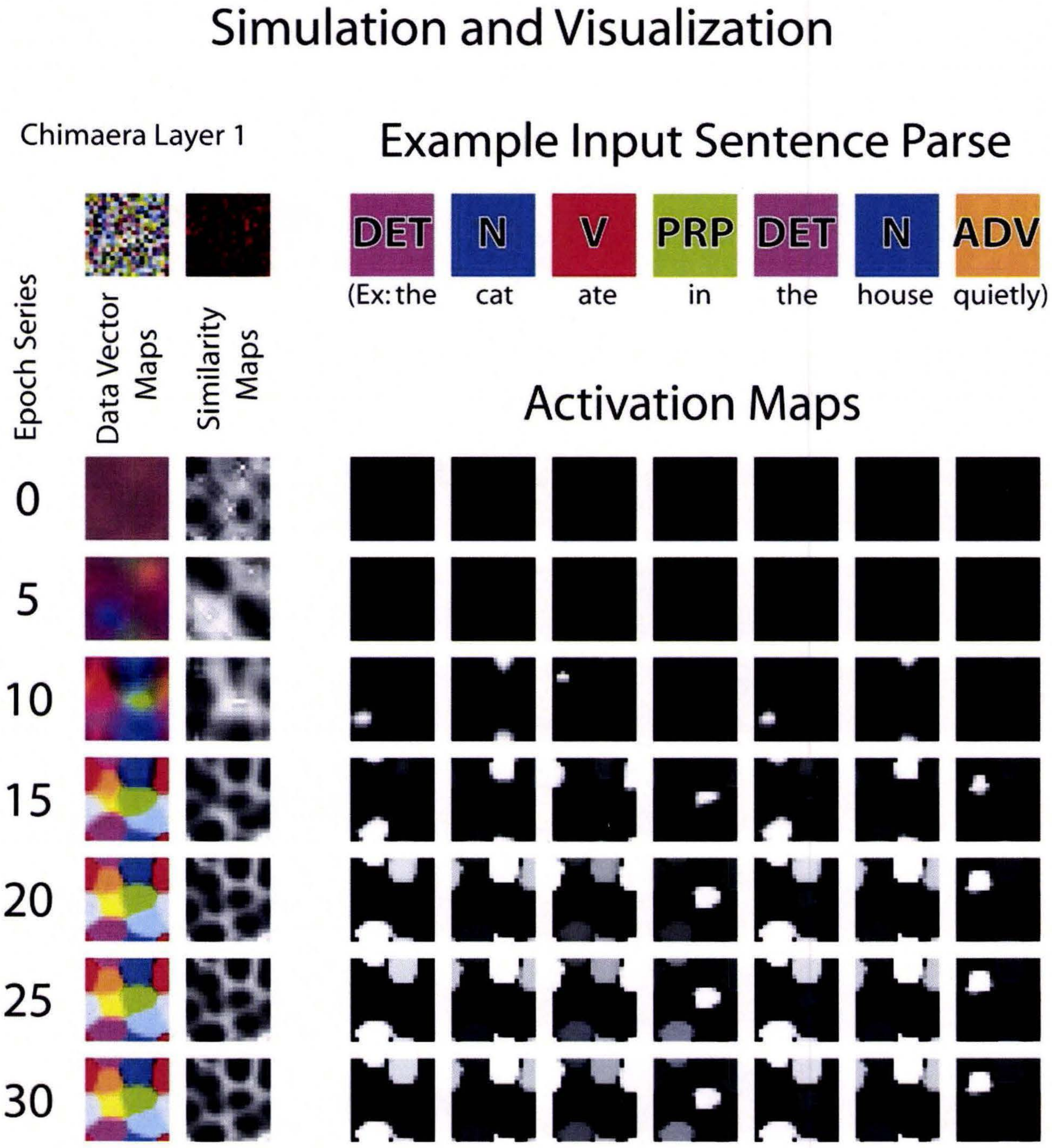

Simulation and Visualization
Chimaera Layer 1
Example Input Sentence Parse
DET
N
V
PRP
DET
N
ADV
(Ex: the
cat
ate
in
the
house
quietly)
Epoch Series
Data Vector Maps
Similarity Maps
Activation Maps
0
5
10
15
20
25
30

Figure 3.2

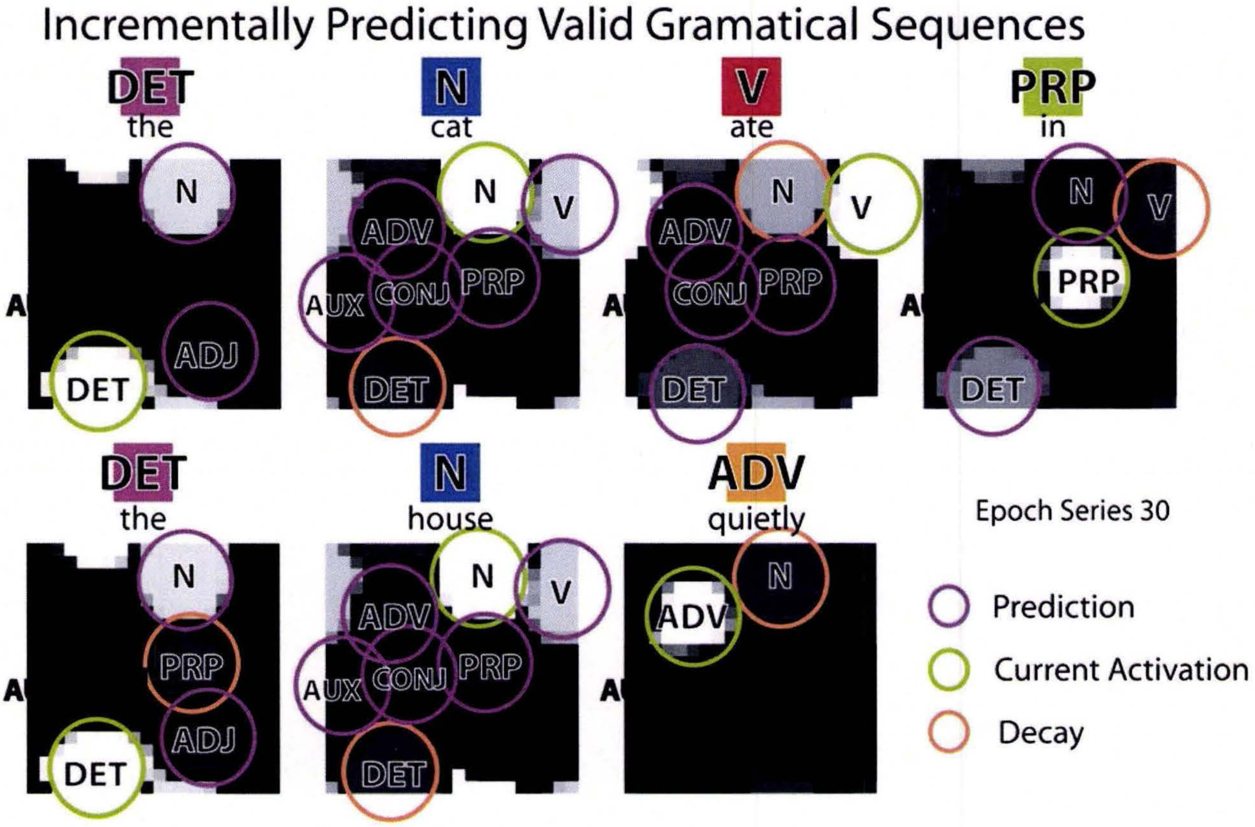

Incrementally Predicting Valid Gramatical Sequences
DET
the
N
ADJ
DET
N
cat
N
V
ADV
AUX
CONJ
PRP
DET
V
ate
N
V
ADV
CONJ
PRP
DET
PRP
in
N
V
PRP
DET
DET
the
N
PRP
ADJ
DET
N
house
N
V
ADV
AUX
CONJ
PRP
DET
ADV
quietly
N
ADV
Epoch Series 30
Prediction
Current Activation
Decay

Figure 3.3

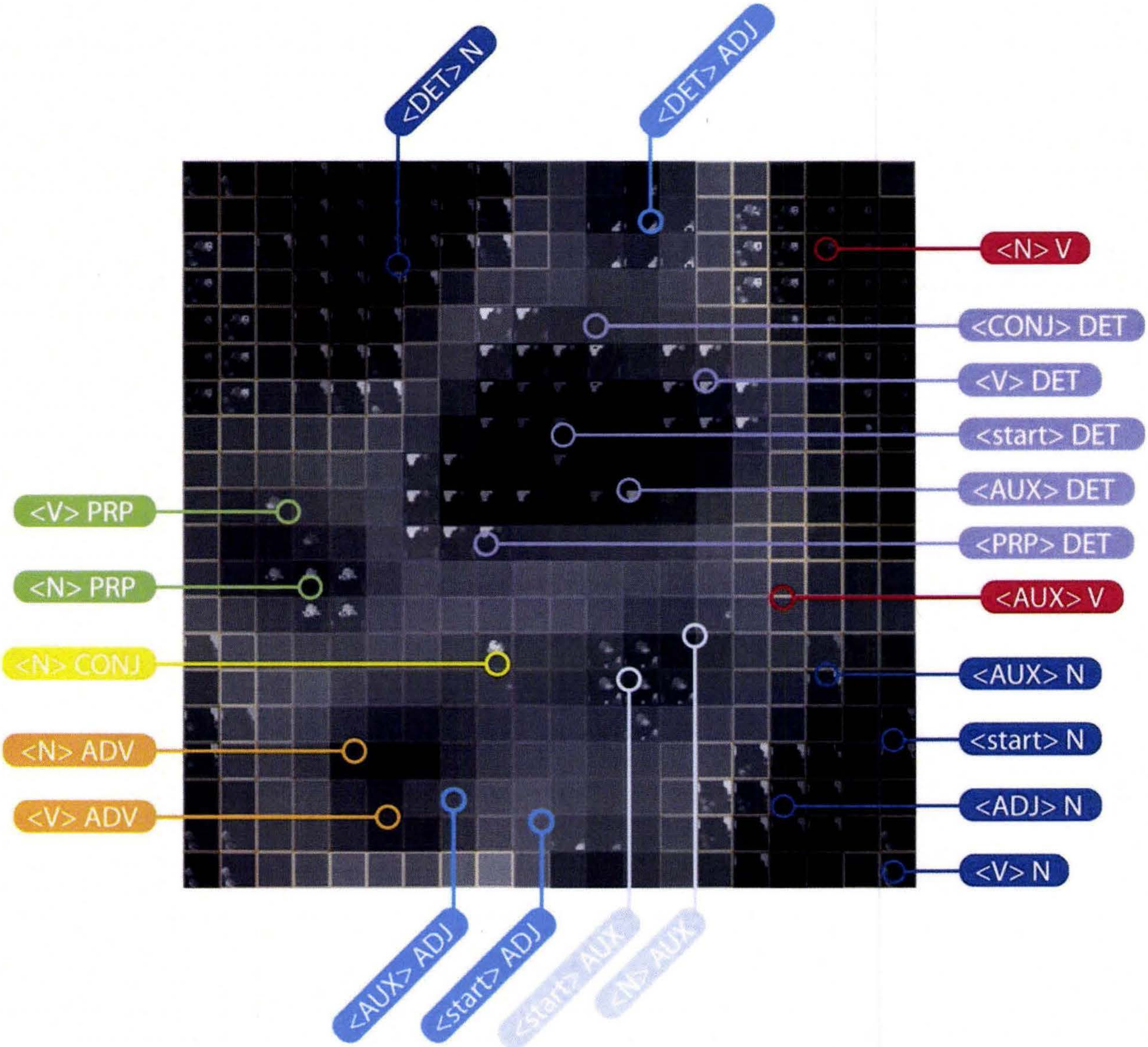

<DET> N
<DET> ADJ
<N> V
<CONJ> DET
<V> DET
<start> DET
<AUX> DET
<PRP> DET
<AUX> V
<AUX> N
<start> N
<ADJ> N
<V> N
<V> PRP
<N> PRP
<N> CONJ
<N> ADV
<V> ADV
<AUX> ADJ
<start> ADJ
<start> AUX
<N> AUX

Figure 3.4

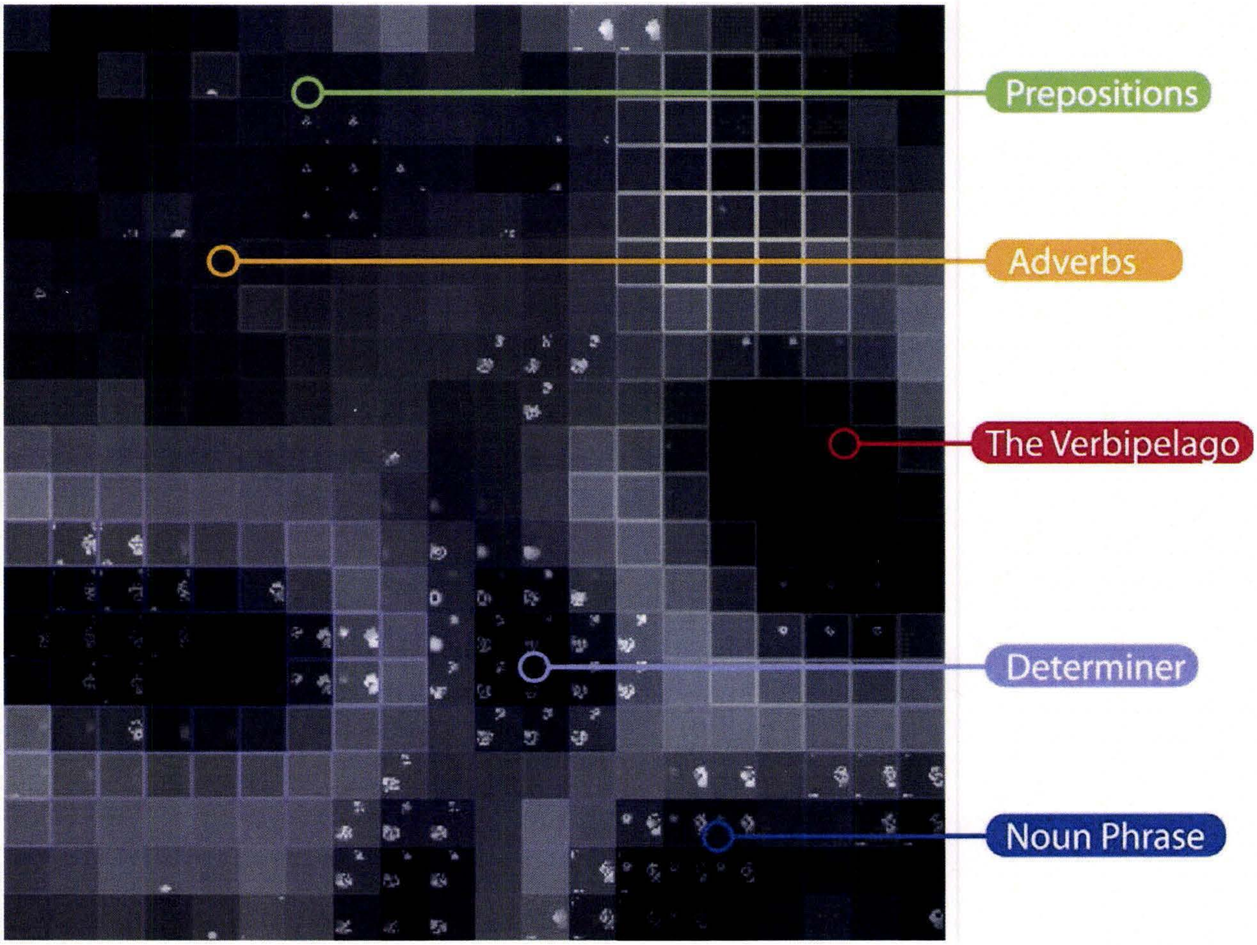

Prepositions
Adverbs
The Verbipelago
Determiner
Noun Phrase

Figure 3.5

Chapter 4

Artificial grammar learning in the simple recurrent network:

An analysis of the performance of supervised architectures on ambiguous grammars

4.1 Introduction

One of the most captivating areas of study in artificial neural network research is the investigation of how connectionist systems can be applied to the problem of learning language. This problem is so large that it can broken into many sub-domains that concern specific aspects of language – for example, learning the phonology (e.g. Kohonen, 1988) or orthography (e.g. Mayraz and Hinton, 2002) of a given language, or the combinations of graphemes and phonemes that compose a given word (e.g. McClelland and Rumelhart, 1981). The formal study of natural language processing generally divides the problem of learning language into two broad categories: learning a low-level transduction of some spoken or written signal into discrete units, of which the study of phonology and orthography are primarily concerned; and acquiring higher-level representations of combinations of these discrete symbolic units to form words and sequences of words, each with some meaning, of which the study of morphology, semantics, and grammar are concerned.

Abstractly, the general problem of learning a grammar is very similar to the problem of learning a potentially complex sequence of instructions composing a computer program, with the necessity to remember certain values, count, or jump to different parts of the grammar. A more restricted subset of grammar learning can be captured in a Finite State Automation (FSA), which is a computational abstraction of a graph where the nodes of the graph represent arbitrary states, and the links between nodes represent the possible allowable transitions from one state to another. A simple grammar that accepts sequences of the form A – B – C – D or A – E – F – D is shown in Figure 4.1.

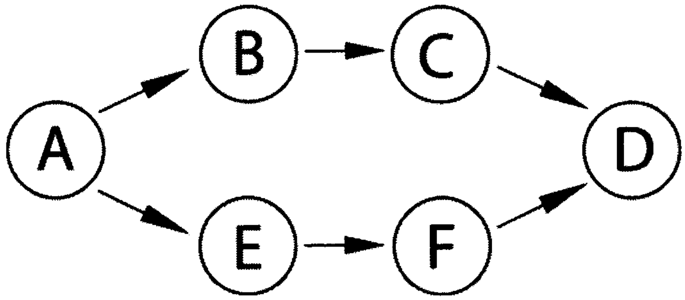


Figure 4.1

While Finite State Automations capture only a subset of a general grammar, they can contain aspects that add computational complexity. For example, consider the case of a grammar where the strings A – X – B and C – X – D are taken to be grammatical, but the sequences A – X – D and C – X – B are ungrammatical. In this case, while at a state that represents a transition to X, we must know the element that preceded X before we are able to determine a possible grammatical transition from X. This manifests itself in the graph as depicted in Figure 4.2, where the same string element (here, "X") will be accepted as input along multiple paths in a finite state automation.

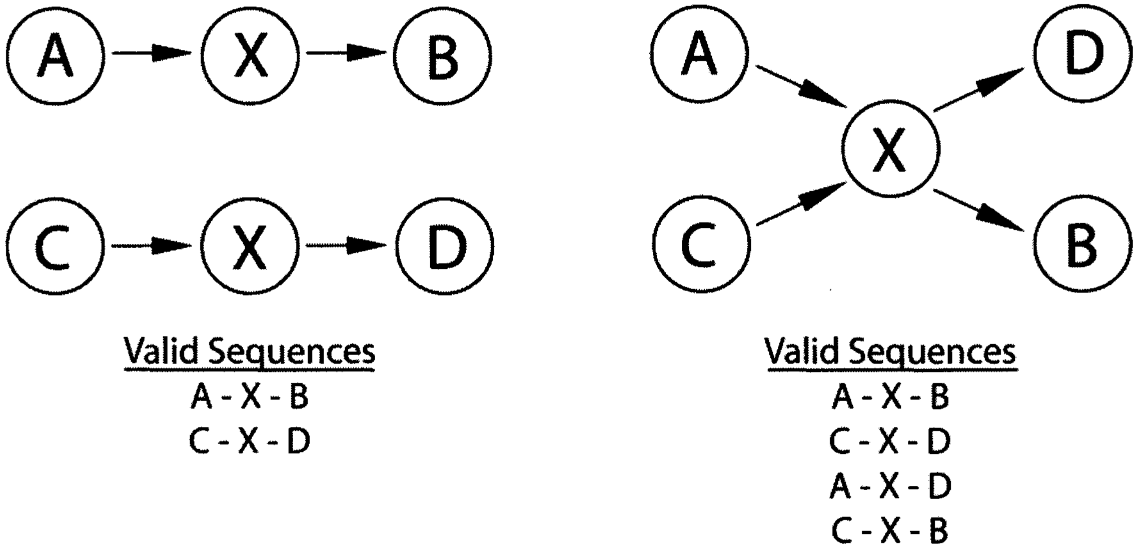


Figure 4.2

From a computational perspective, the requirement to know not just where we are in a string, but also *where we have been*, adds significant complexity. In the case of a simple grammar that does not contain a "transitional ambiguity", the knowledge of where we are in a parse can be maintained simply through a knowledge of the current single element we are processing in a string. However, in the case of a transitional ambiguity, the system requires a mechanism to "look-back" a certain number of elements to determine the next possible grammatical transition. In the general case, for an "n-back" transitional ambiguity, this mechanism would have to look back *n* elements in the string in order to resolve this ambiguity.

Here we examine the processing and performance capabilities of two supervised neural network architectures as they acquire "n-back" artificial grammars that vary in complexity, where "n-back" grammars were selected as they were identified as particularly difficult for the unsupervised, feed-forward, and self-organizing Chimaera architecture (described in Chapter 2) to acquire – generally requiring $n+1$ feed-forward layers to successfully represent an *n-back* grammar. Where Chapter 3 identified recurrence as a likely means of efficiently representing deeply ambiguous transitions, here we experimentally investigate the performance of both a 3-layer feed-forward network (Rumelhart et al., 1986), as well as the Simple Recurrent Network (SRN; Elman, 1990) (which extends the 3-layer feed forward architecture through the addition of a recurrent temporal processing context layer) on an abstract grammar learning task with varied depths of grammatical ambiguity.

4.2 Methods

4.2.1 3-layer feed forward network

The 3-layer feed forward architecture consists of a three layer network of nodes, where a given layer is fully connected to its respective superordinate and/or subordinate layer(s). The flow of activation proceeds from an input layer, through a hidden layer, then finally to a layer of output nodes. The weights between these nodes are trained using the backpropagation algorithm, which extends the delta rule to propagate error terms backward across multiple layers and allow the training of deep multilayer networks. At a given epoch, a training pattern is presented to the network (for example, for the sequence A – X – B, the input pattern corresponding to "A" would first be presented), while the output nodes are clamped to a pattern corresponding to the *(n+1)*th element in the sequence (in this case, "X"). The weights in the network are then adjusted, where the delta values for the output and hidden layers are first calculated:

$$\delta_k = o_k(1-o_k)\cdot(o_k - d_k) \tag{1}$$

$$\delta_h = o_i(1-o_i)\cdot\sum_k (w_{kh}\cdot\delta_k) \tag{2}$$

and used to calculate the change in weight for each node:

$$\Delta w_{ji} = -\varepsilon\cdot\delta_j\cdot s_j \tag{3}$$

where epsilon (ε) represents a learning rate, and a sigmoidal activation function is used to determine the activation of a given node:

$$s_j = sigmoid\left(\sum_i w_{ji}\cdot s_i\right) \tag{4}$$

$$sigmoid(x) = \frac{1}{1+e^{-x}} \qquad (5)$$

Training continues until the network reaches a steady-state, where the error is either near-zero or has converged upon a value and is no longer significantly decreasing.

4.2.2 Simple Recurrent Network

The simple recurrent network extends the 3-layer feed forward network by adding a layer of recurrent "context" nodes that carry information about what the network has recently processed. The content of these nodes are the activation values of the hidden layer at the previous epoch, and this serves as partial input to the hidden layer at the current epoch. This mechanism functionally resembles an input layer that is divided into two sections – one that contains the input vector representing a given element in a sequence, and another that directly mirrors the content of the hidden nodes from the previous epoch, and supplies input about the temporal "context" of the input vector to the current epoch. This architecture is depicted in Figure 4.3.

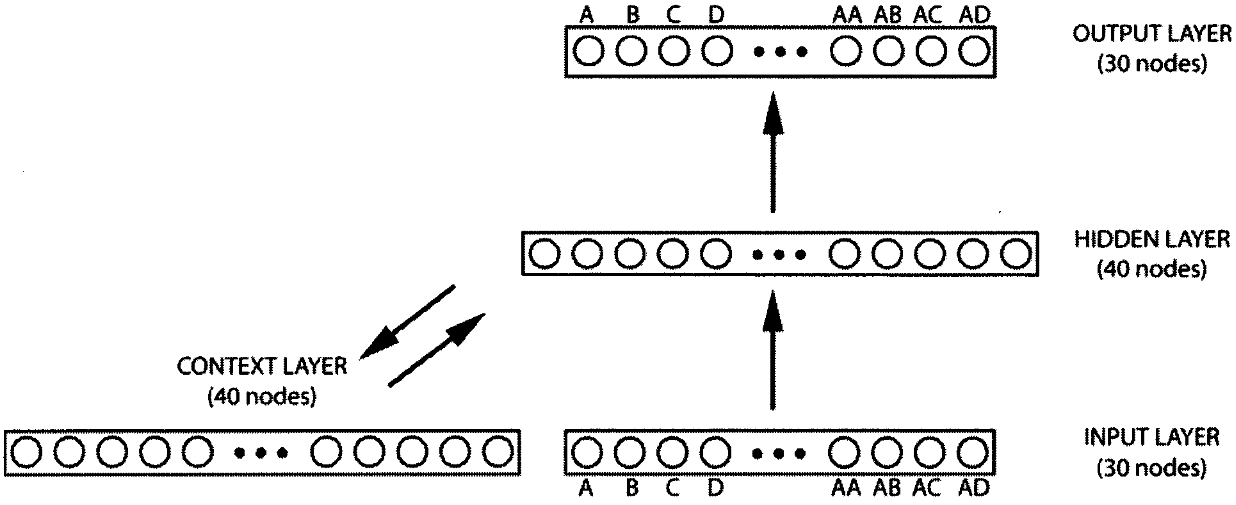


Figure 4.3

4.2.3 Input and Output Representations

The nodes at both the input and output layers represent individual sequence elements ("A", "B", "C", etc.). For simplicity, these are localist representations, where the input or output pattern for a given element contains only a single non-zero element, set to one, and as such each input vector is unique to a given sequence element.

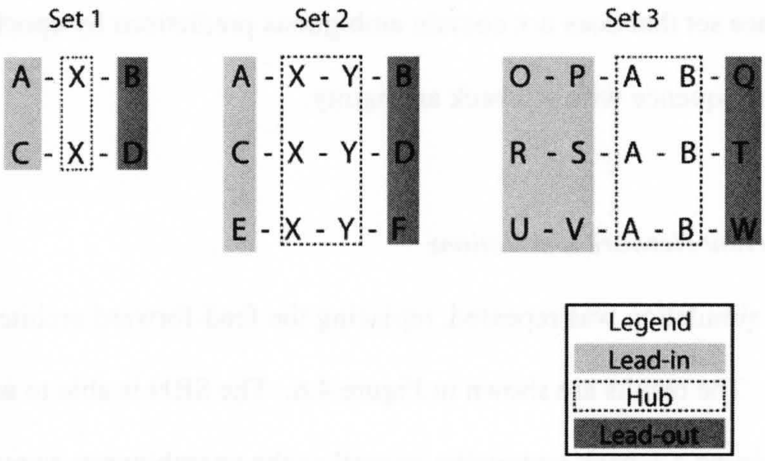


Figure 4.4

4.2.4 Sequence Set

The input sequence set used in simulations with both the feed-forward and recurrent architectures was varied along several dimensions. Figure 4.4 describes several dimensions of an n-back sequence: (1) the depth of the ambiguity, which can also be thought of as how many items one must look back in order to resolve the ambiguity, referred to here as the *"hub size"*, (2) the *"lead-in"*, or number of unique and non-ambiguous sequence elements that precede the transitional ambiguity, (3) the *"lead-out"*, or number of unique and non-ambiguous sequence elements that follow the transitional ambiguity, (4) the length of each sequence, in numbers of elements, and (5) the total number of sequences in a given set.

4.3 Results

4.3.1 3-Layer Feed Forward Simulation

The results of the 3-layer feed forward simulation are displayed in Figure 4.5. In this simulation, the list size was held constant (4 sequences of 3 elements each), and the sequences of each sequence set contained either 1 lead-in, a 0 depth hub, and 2 lead-outs (a non-ambiguous sequence), or one element for each of the lead-in, hub, and lead-out (a sequence with a 1-back transitional ambiguity). The results show that while the three-layer feed forward network is able to learn the sequence set that does not contain ambiguous predictions by epoch 160, the network is unable to learn a sequence with a 1-back ambiguity.

4.3.2 Simple Recurrent Network Simulations

The above simulation was repeated, replacing the feed-forward architecture with a simple recurrent network. The results are shown in Figure 4.6. The SRN is able to acquire both the sequence set containing a 1-back ambiguity, as well as the unambiguous sequence set.

4.3.3 Effect of sequence length and number of sequences learned on training time

The effect of the total number of sequences learned on overall training time is displayed in Figure 4.7, for both ambiguous (bottom) and non-ambiguous (top) sequence sets, while the effect of sequence length on overall training time is displayed in Figure 4.7C. In general, for both sets, as the total number of elements the network must learn increases – either through an increase in the number of sequences, or the number of elements per sequence – so too does the training time.

4.3.4 Effect of deep n-back ambiguities

The effect of deep n-back ambiguities on performance is displayed in Figure 4.8. In this simulation, the set size was held at four, the length of each sequence was held constant at 7, while the n-back ambiguity was varied from between 0 and 5 elements long. While training time and performance characteristics remain constant for hub depths of zero to two, this performance quickly attenuates, taking a greater amount of time to train the network for a hub depth of three, a much greater time for a hub depth of four, and plateauing without successful acquisition at a hub depth of five. To test if this performance deficit was caused by the saturation of the hidden nodes, the hidden layer was increased from 40 to 80 nodes, and these results are displayed in Figure 4.9. Here, performance remains constant for hub sizes between one and four, while decreasing slightly for a hub size of five. Similarly, in Figure 4.9 (bottom), performance progressively attenuates for 6-back and 7-back ambiguities, while the sequence set with an 8-back ambiguity is unable to be learned. This pattern of results is consistent with the notion that the performance deficit is caused by a saturation of the hidden nodes.

4.3.5 Effect of lead-in on training time

The effect of variable-length lead-ins before an ambiguous hub is shown in Figure 4.10. In general, the number of elements preceding an ambiguous transition does not affect performance. A notable exception is the case where there is zero lead in, however conceptually this is not an ambiguous transition as there is no context to determine the correct transition. As such, the case of a zero lead-in is just a regular one-to-many free transition in the grammar, and not an ambiguous transition.

4.4 Discussion

While the 3-layer feed forward network was able to learn sequences without ambiguous transitions, it was unable to acquire representations of a sequence set with a 1-back ambiguous transition. Intuitively, this result makes some sense – learning a set of sequences where a given sequence element is unique and unambiguously predicts another element can essentially be reduced to the task of learning a set of unique stimulus-response pairs, whereby for a given element one always gives a corresponding output. However, when the network requires additional information in order to predict the next item – for instance, requiring the previous item in the sequence – in the absence of a mechanism to maintain such a representation, the network lacks the computational capabilities to resolve the ambiguities in the sequence set, and will deliver an ambiguous prediction.

The simple recurrent network performed very well on learning grammars with n-back ambiguities, with the only significant effect on performance arising from the depth of the ambiguity. Examining the performance of individual sequence elements across all sequences, Figure 4.11 (top) shows that the SRN generally learns the lead-in quickly, as well as the elements of the hub leading up to the final hub element and ambiguous transition. Learning this transition is much slower, and generally occurs long after all other transitions – both preceding and following the ambiguous transition – are learned. The mechanism for this slow but eventual learning is likely that the hidden layer is acquiring separate and distinct representations for each state in the finite state automation representing the n-back grammar, and that learning these "sequence element plus temporal context" paired representations can take more time than simply the sequence element itself (Servan-Schriber, Cleeremans, and McClelland, 1989), especially in the presence of limited hidden unit resources.

In summary, a recurrent network supplies a temporal context as input that an otherwise feed-forward network can use to successfully learn complex abstract grammars and ambiguous sequence sets, even those with deep n-back ambiguities. The ability to learn general grammars and finite state automata has important implications for the processing capabilities of neural networks, and is an important requirement in learning high-level representations of the structure of language, which depend on complex grammars that combine discrete symbols into larger units of meaning across various levels of representation. This examination of the acquisition of deeply ambiguous grammars in a recurrent supervised architecture has determined that recurrence provides a mechanism to resolve the precise ambiguities that the unsupervised Chimaera architecture of Chapter 2 has particular difficulty with. As such, and following Chapter 3, extending the Chimaera architecture and its correlational-based temporal learning with a mechanism of temporal recurrence should enable networks with comparatively few layers to have the capacity for efficiently acquiring deeply ambiguous grammars.

Appendix

Unless otherwise noted, the learning rate epsilon (ε) was set to 0.2 for all simulations. The input and output layers contained equal numbers of nodes, generally between 30-50, and this number was held constant across simulations comparing the effect of a given sequence aspect on network performance. Unused input vector elements were set to zero, and had no influence on further processing. Across all simulations, the size of the hidden layer was 40 nodes, unless otherwise noted. Weights were initialized to random values between +/- 1.0 before training. Because of this non-deterministic element, performance is generally compared across the average of multiple simulations, generally no less than 10.

Figure Captions

Figure 4.1: An example finite state automation.

Figure 4.2: Two example finite state automations illustrating ambiguous (left) and unambiguous (right) transitions.

Figure 4.3: A schematic of the simple recurrent architecture used in the simulations. Input to the hidden layer consists of both "input" and "context" nodes, where the values of the hidden layer on a given epoch are copied and used as context input on the following epoch. Representation in the input and output layers is localist, with only a single element active on a given epoch.

Figure 4.4: Several examples of ambiguous sequence sets, illustrating the properties of lead-in, hub depth, and lead-out.

Figure 4.5: The performance of a 3-layer feedforward network on an ambiguous (hub depth 1) and unambiguous (hub depth 0) sequence.

Figure 4.6: The performance of a simple recurrent network on an ambiguous (hub depth 1) and unambiguous (hub depth 0) sequence.

Figure 4.7: The effect on training time of (A) unambiguous, (B) ambiguous (hub depth 1), and (C) length on total training time.

Figure 4.8: The effect of deep ambiguities on sequence training time, for ambiguities varying from 0 to 5 elements long. (Hidden node size = 40)

Figure 4.9: The effect of deep ambiguities on sequence training time, for ambiguities varying from (top) 0 to 5 elements long, and (bottom) 6 to 8 elements long. (Hidden node size = 80)

Figure 4.10: The effect of variable length lead-ins on sequence training time, for lead-in lengths between 0 and 5 elements long.

Figure 4.11: Element-specific sequence performance in the case of sequences with one (top) and many (bottom) elements following the ambiguous transition. For both sequence sets (top and bottom), the red line (SE3) represents the last hub element, directly before the 2-back ambiguous transition.

Ambiguous and Unambiguous Sequence Learning in a 3-layer Feed Forward Network

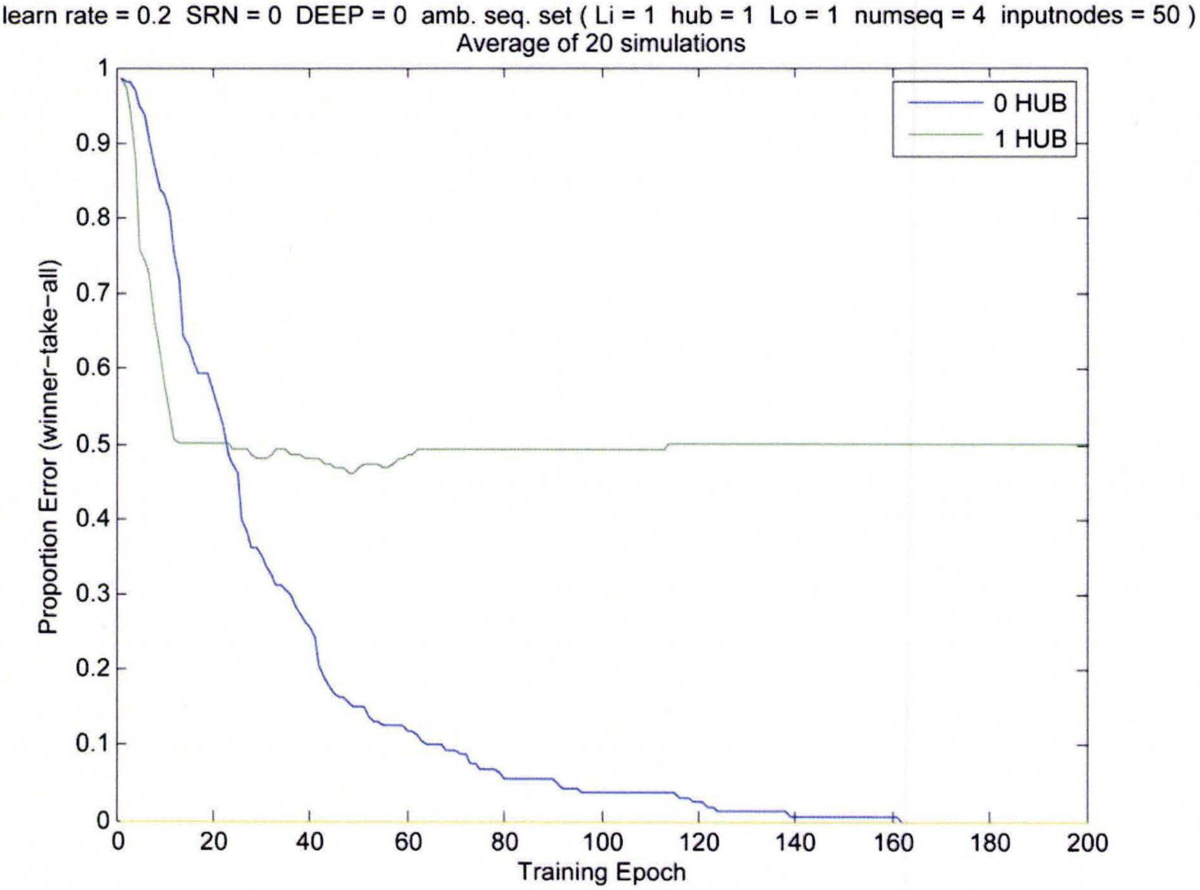


Figure 4.5

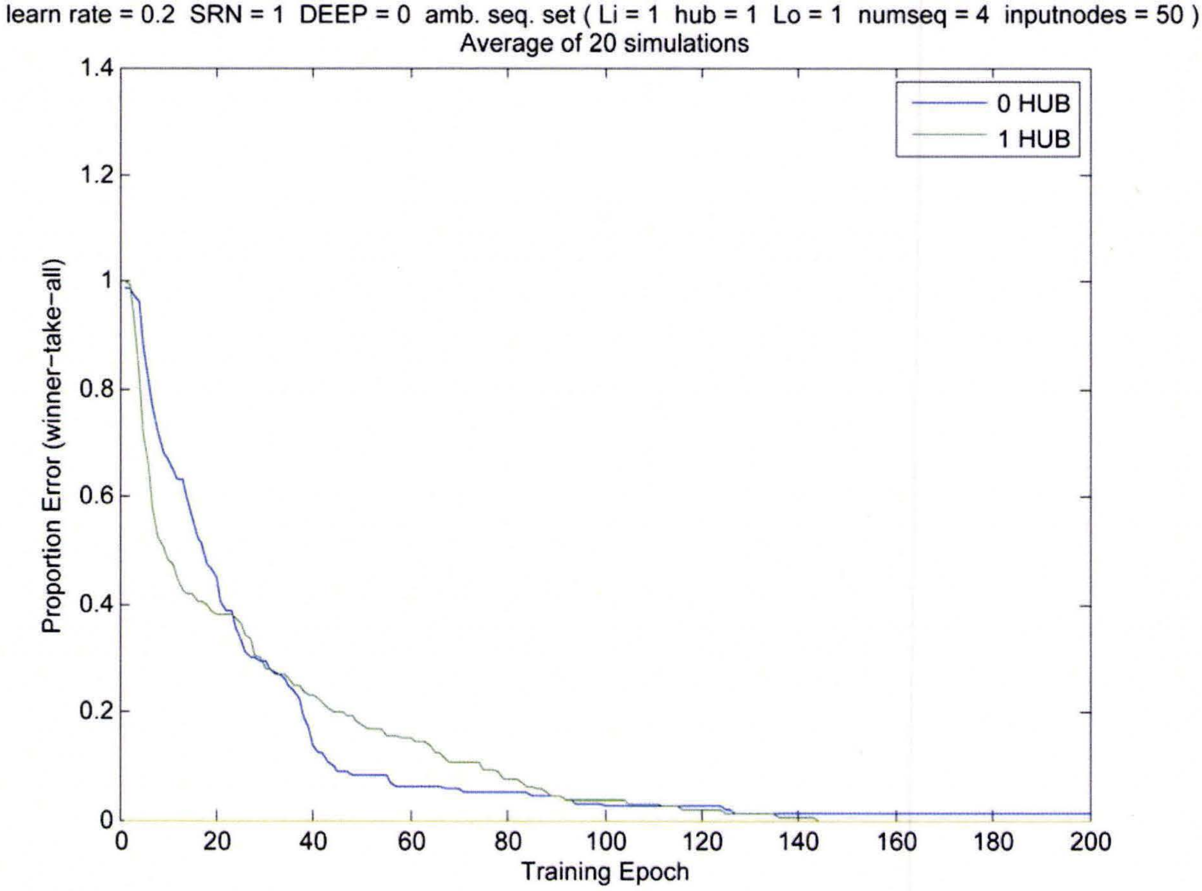
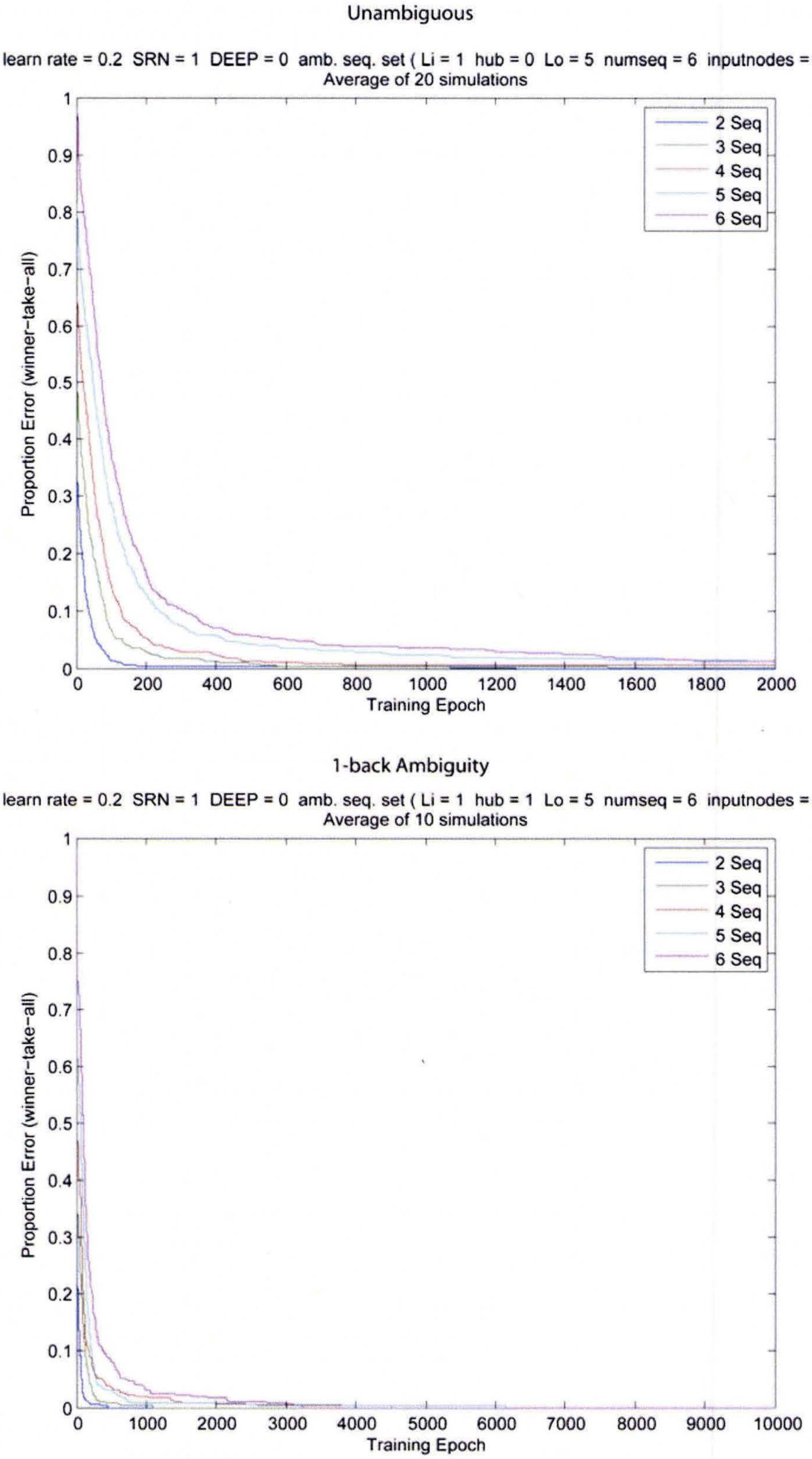

learn rate = 0.2 SRN = 1 DEEP = 0 amb. seq. set (Li = 1 hub = 1 Lo = 1 numseq = 4 inputnodes = 50)
Average of 20 simulations
0 HUB
1 HUB
Proportion Error (winner−take−all)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
20
40
60
80
100
120
140
160
180
200
Training Epoch

Figure 4.6

Effect of Number of Sequences Learned on Training Performance



Unambiguous

1-back Ambiguity

Figure 4.7

Effect of Sequences Length on Training Performance

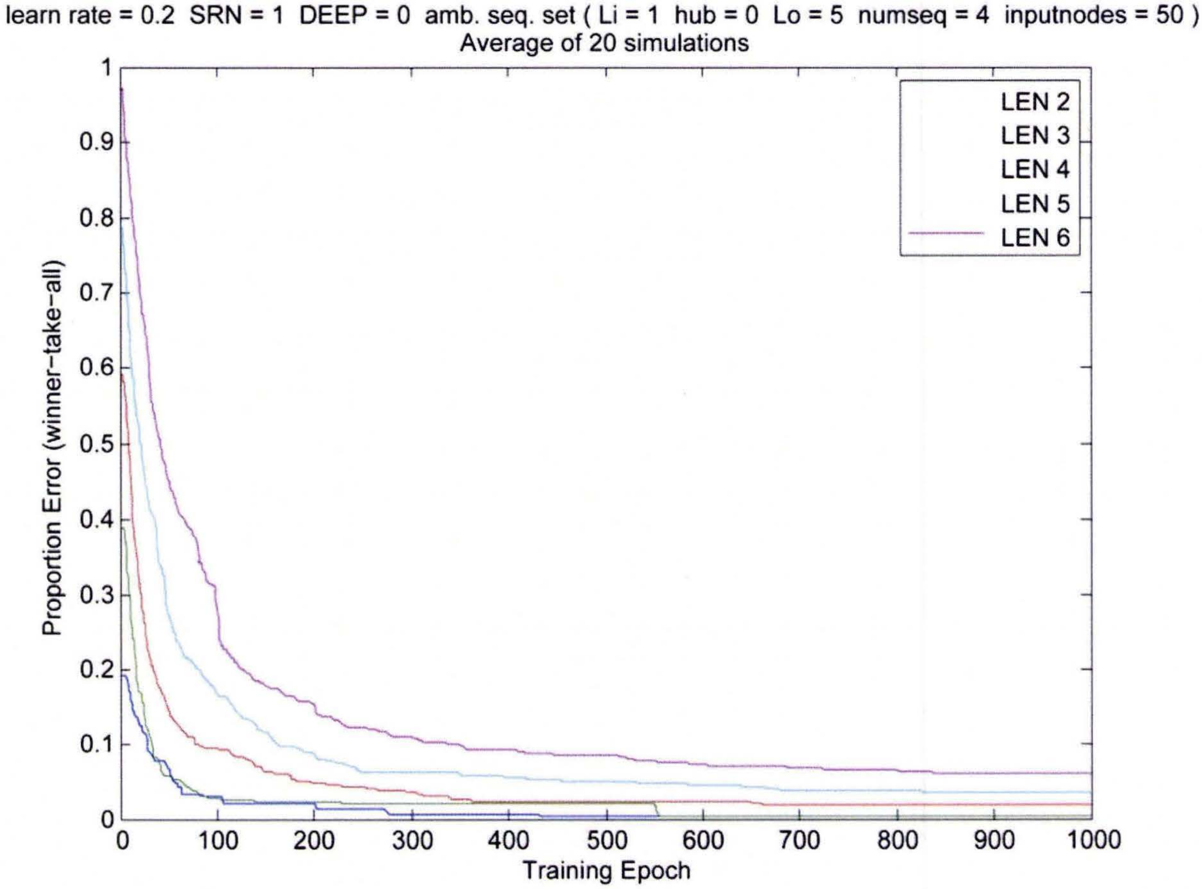


Figure 4.7C

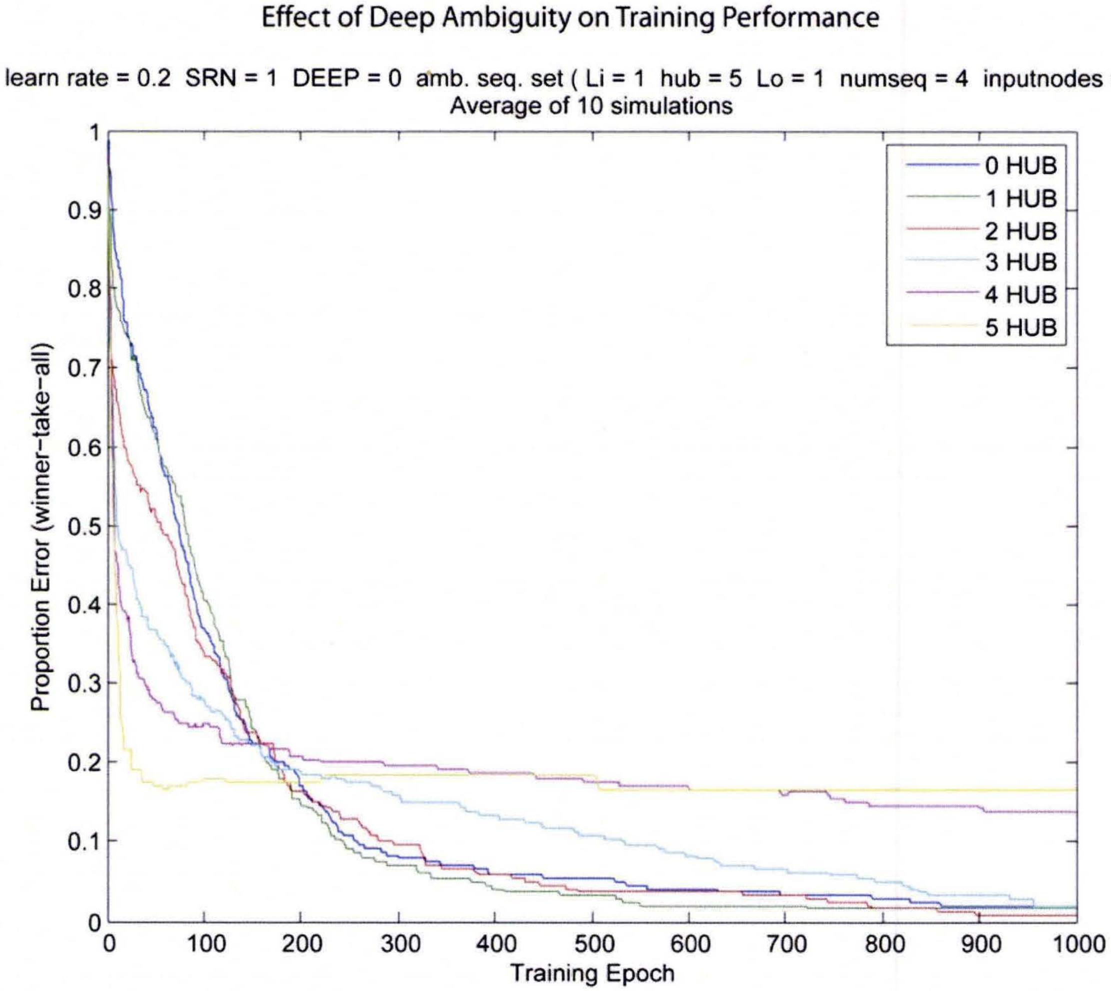

Effect of Deep Ambiguity on Training Performance
learn rate = 0.2 SRN = 1 DEEP = 0 amb. seq. set (Li = 1 hub = 5 Lo = 1 numseq = 4 inputnodes = 40)
Average of 10 simulations
Proportion Error (winner−take−all)
Training Epoch
0 HUB
1 HUB
2 HUB
3 HUB
4 HUB
5 HUB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
100
200
300
400
500
600
700
800
900
1000

Figure 4.8

Effect of Deep Ambiguity on Training Performance

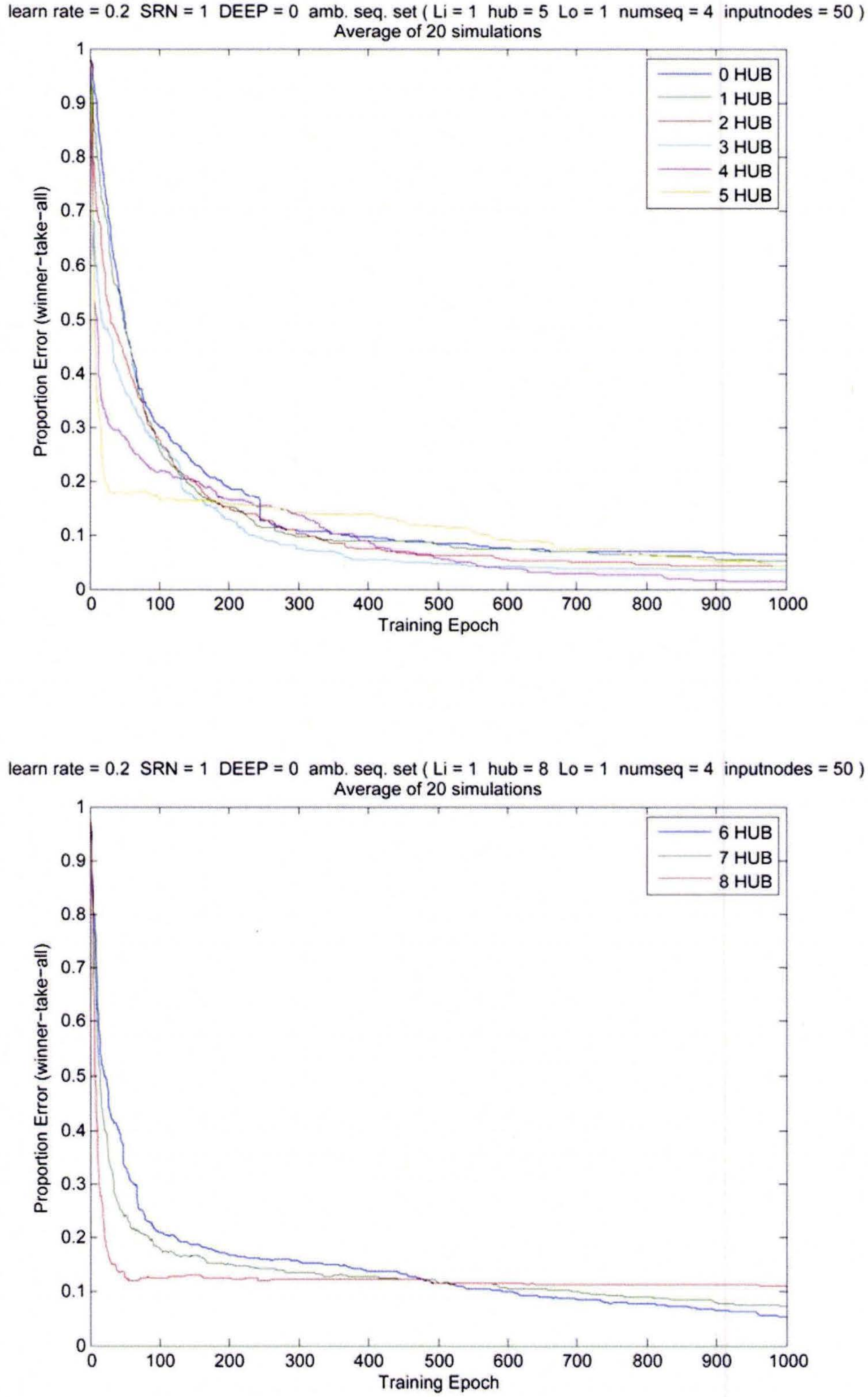


Figure 4.9

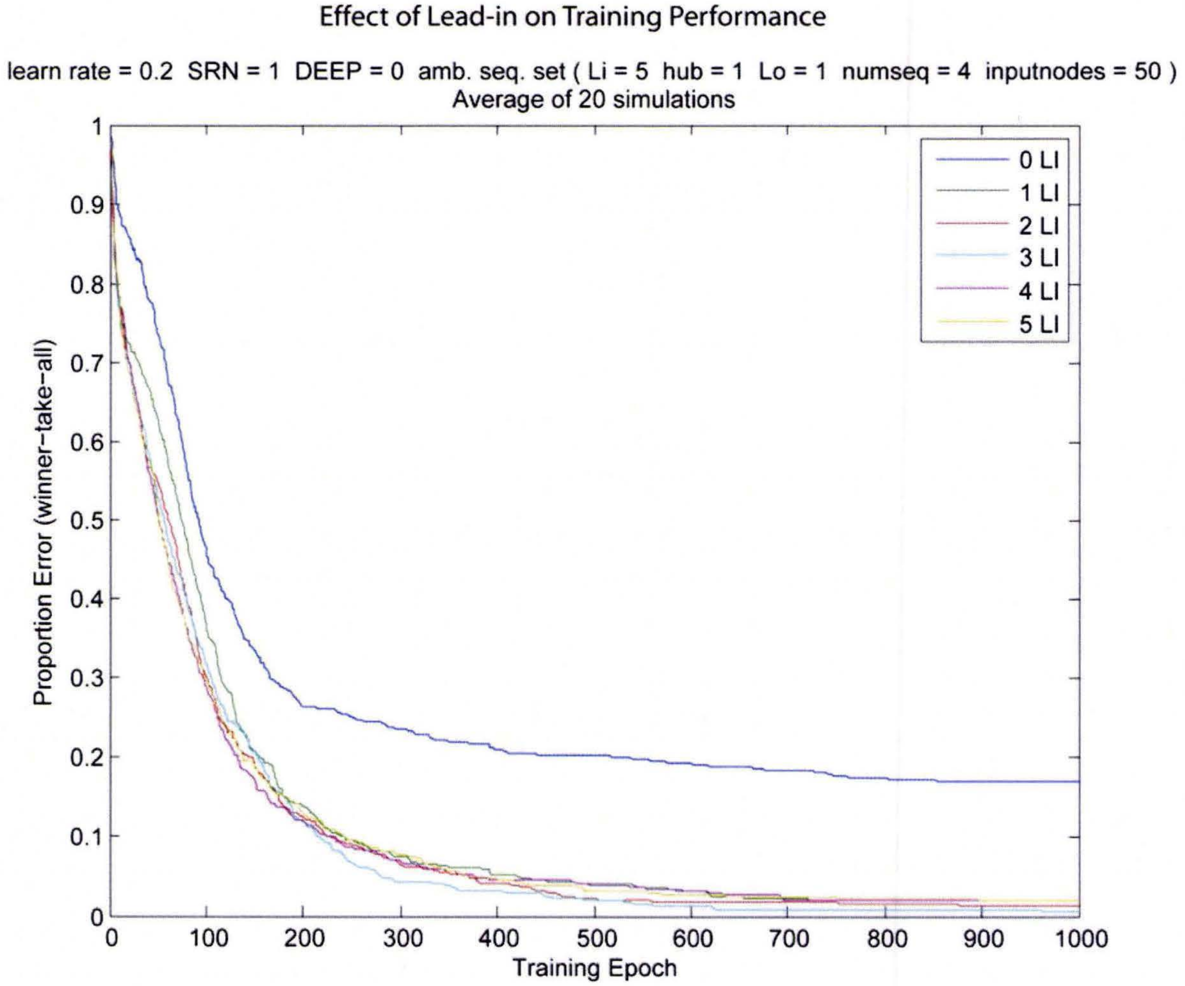
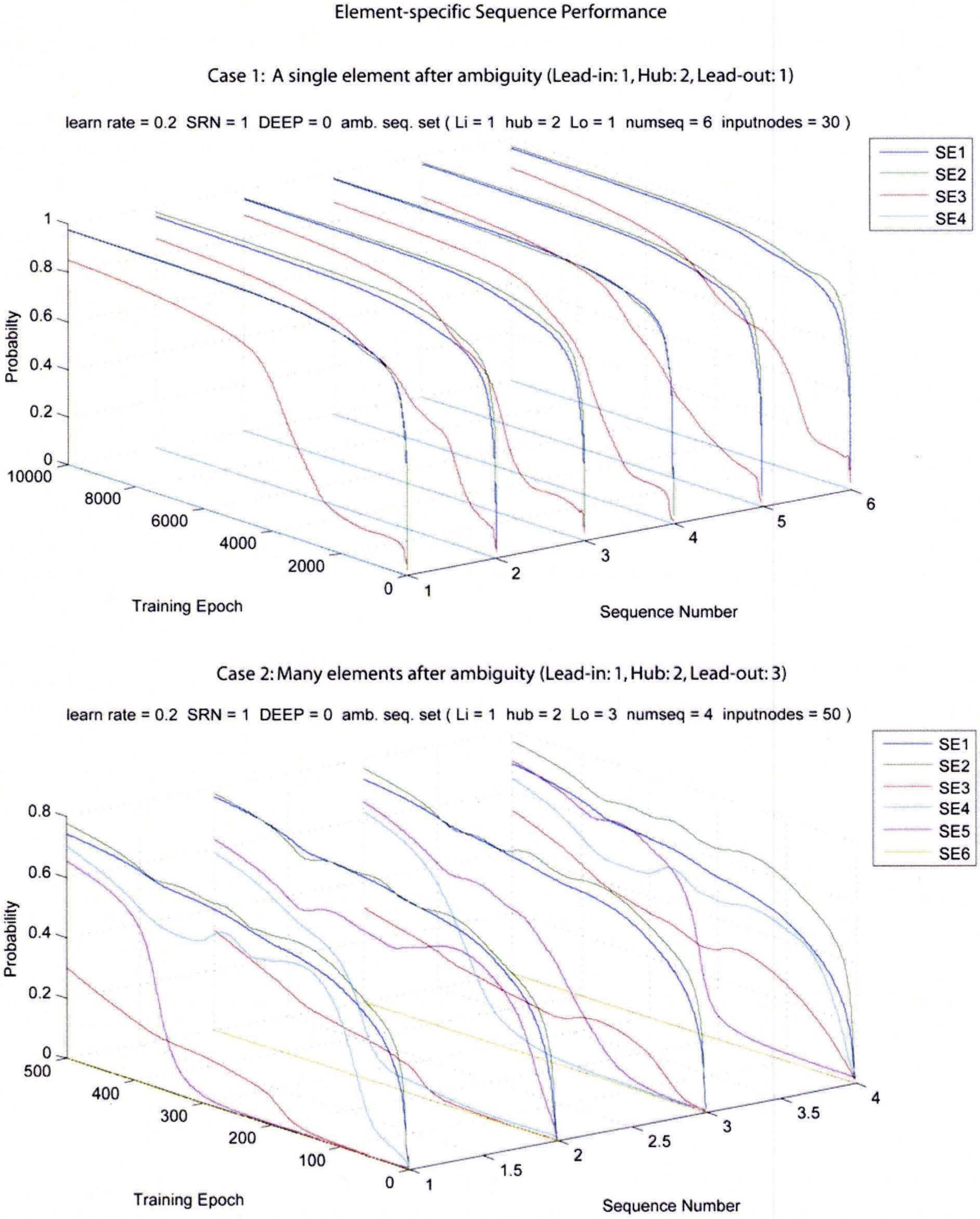

Effect of Lead-in on Training Performance
learn rate = 0.2 SRN = 1 DEEP = 0 amb. seq. set (Li = 5 hub = 1 Lo = 1 numseq = 4 inputnodes = 50)
Average of 20 simulations
Proportion Error (winner-take-all)
Training Epoch
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
100
200
300
400
500
600
700
800
900
1000
0 LI
1 LI
2 LI
3 LI
4 LI
5 LI

Figure 4.10



Element-specific Sequence Performance

Case 1: A single element after ambiguity (Lead-in: 1, Hub: 2, Lead-out: 1)

Case 2: Many elements after ambiguity (Lead-in: 1, Hub: 2, Lead-out: 3)

Figure 4.11

Chapter 5

Recurrent Chimaera

5.1 Introduction

Chapter 3 introduced a feed-forward multi-layer Chimaera network for the processing of an unambiguous simple grammar containing 8 parts of speech and 20 sentence structures, while critically identifying the requirement for the Chimaera to move from a feed-forward intra-layer recurrence architecture toward an architecture incorporating an inter-layer recurrence mechanism capable of deep look-back for transitional ambiguity resolution.

Chapter 4 examined the acquisition of abstract grammars using the supervised simple recurrent network architecture, experimentally verifying that the recurrent architecture of the SRN is broadly capable of processing ambiguous sequences. Given assumptions of large numbers of hidden units and long periods of training, Chapter 4 showed the SRN is able to store sequence information in abstract transitionally-ambiguous grammars that contain many of the aspects of linguistic grammars, including varied numbers of context elements preceding an ambiguous prediction, as well as potentially deep ambiguities requiring a mechanism of lookback far more elements deep than the network has layers. The SRN and its mechanism of recurrence are able to successfully acquire a diverse set of these sequences, with little effect of the actual content or depth of ambiguity in the sequences themselves.

From a computational perspective, each layer of a feed-forward network can be thought of as capable of performing only a single processing "step" in a given computer program. As such, iterating through multiple processing steps requires either many layers, or a mechanism to feed back the processing of a later layer into an earlier layer, and use this input to supply information required for processing. Computationally, this mechanism of recurrence provides the capability of a loop, allowing (in principle) as many different operations and states as the network can pack into the representational and transduction capabilities of the recurrent layers, for a given number of recurrent nodes, and a given learning algorithm's capability of modifying the weights of the recurrent layer such that meaningful processing will occur.

Many unsupervised recurrent architectures based on the self-organizing map have been proposed, with a particular emphasis on providing temporal extensions to the SOM algorithm (eg. TKM, Chappell and Taylor, 1993; RecSOM, Voegtlin, 2002) or with a particular sensitivity to clustered data (e.g. trees, SOMSD, Hagenbuchner et al., 2003; MergeSOM, Strickert and Hammer, 2003). Chapter 4 described how recurrence is approached using a simple recurrent network using the supervised backpropagation algorithm – typically through a mechanism of "copying" the activation pattern of a superordinate layer at a given epoch, and using this as partial, contextual input to a subordinate layer at the next epoch. In some cases this contextual input may itself be further transduced through trained (or, in cases, even sparse, untrained) connections that allow extra processing steps to take place (e.g. ESN, Jaeger, 2003). Unsupervised architectures tend to take a similar approach to recurrence, where the overarching mechanisms of recurrence are the same as in supervised networks, such that the difference tends to remain in how the learning algorithm processes this new information available to the network.

As self-organizing maps are themselves a high-level abstraction of a low-level learning rule that produces the characteristic topographic organization of similar representations over perceptual cortex, recurrent self-organizing maps tend to lack a concept of explicit activation, and instead are required to use alternate information for recurrence. SOMSD is one example of a recurrent self-organizing map, where the network uses the spatial coordinates of the best matching node to the input vector at a given epoch as recurrent contextual input at the next epoch. In this case, because the spatial coordinates of the best matching node may contain far fewer dimensions than the input vector, both the normal and recurrent input have ascribed weights to balance the contribution of each of the normal and recurrent input vectors, such that each may be considered more or less heavily in determining the location of the next best-matching node.

5.1.1 Temporal Representation in the Chimaera Network

Approaching the issue of recurrence much closer to a simple recurrent network than a self-organizing map, the Chimaera architecture described in Chapter 2 is conceptually much closer to a cellular automation (as described by, for example, Wolfram, 2002), and generates an activation map containing the activation level of each node in a given layer, for each layer in the network. Alone, where the information in this activation map includes only the activation produced as a result of a given input vector, this activation map is nearly functionally equivalent to the mechanism used by the SOMSD architecture – it will generally contain the best-matching node (which will be the most active node in the activation map), as well as a cluster of similar nodes surrounding that best-matching node, the size and activation of which will be determined by the relative packing density of the network. In the extreme case, layers with many representations and few nodes will contain a small number of active nodes in the input network, where this will approach only a single active node in the case where the number of nodes in the network and the number of unique dissimilar representations in the network are identical.

At the other extreme, where there are relatively plentiful numbers of nodes for each unique input vector the network is to store, the activation map will contain a broad region of active nodes in response to a given input vector. By comparison, in terms of similarity, the activation function of the Chimaera can be modulated such that similar vectors (shades of a colour, for instance) produce a non-zero level of activation, where the colour that best matches the input vector will produce the highest activation, and each shade will produce proportionally less activation. In this way, recurrent information stored in the activation map also has the potential to explicitly carry information about similarity – at least, more information about similarity than a method where one has access to only a best-matching spatial location, where a learning rule using that recurrent information must then make an inference of similarity based on spatial topography. While it is the case that the SOM learning rule generally organizes input vectors based on similarity, unless the data being presented is equidistant in the n-dimensional input space being represented, some of the input vectors are going to be more or less similar than others. Intuitively, this makes sense – the similarity map visualization is not homogeneous, but rather the border regions between representations are lighter or darker depending on the relative similarity between regions. Some nearby regions may be very similar – and are in fact *likely* to be so, given the clustering nature of the SOM learning algorithm – but in the extreme yet illustrative example, were a sample input set to include 100 vectors representing similar shades of bright red, and a single vector representing blue, that blue vector would still have to be stored *somewhere* in the network, and that spatial location is going to be spatially near other regions. While spatial topography conveys information about similarity in general, this is not always the case. The activation map of the Chimaera affords the potential to preserve at least some of this similarity information over best-matching unit approaches, and we will examine how this is useful to grammar learning in the next chapter.

While we have examined two types of information the activation map can convey – the best-matching node of the current element in a sequence, as well as similarity information – the activation map can contain other information as well. One critical type we have already been including is the network's *processing context* – information that includes not only the current element of a given sequence the network is traversing, but also, in a SARDNET like fashion (James and Miikkulainen, 1995), the previous element the network visited, as well as possible predictions for the next element. This intra-layer recurrence contains a wealth of sequence processing information, and is the mechanism of temporal sequence prediction that the feed-forward Chimaera has used. In the case of a recurrent architecture, this information is already explicitly provided by virtue of a recurrent layer having access not only to input representing the current sequence element, but also to recurrent activation representing the previous element in the sequence. Where temporal information is explicitly fed back into the network as an input vector, the network is then able to make use of the SOM learning algorithm to create an explicit representation of a given node in a given temporal context, and does not require a carefully balanced system of activation and intra-layer association and flow to represent temporal sequences. As such, this intra-layer information is potentially redundant, and may not add any significant processing benefit when inter-layer recurrence is substituted for the previous intra-layer recurrence, except in the case of the final (output) layer of a network, where this intra-layer recurrence mechanism provides a facility for the generation of temporal predictions.

In our design of a recurrent Chimaera, we can infer that the activation map that has previously served as feed-forward input to subordinate layers of a feed-forward Chimaera network may similarly serve as recurrent contextual input to a recurrent Chimaera, very similar to the specification of a RecSOM (Voegtlin, 2002) with a Chimaera-like energy function. The information contained within this temporally-delayed activation map is – in the worst case – functionally equivalent to using the spatial location of the best-matching node as recurrent information, while also offering potential mechanisms to convey a knowledge of the similarity between representations, as well as information that describes the temporal activation of multiple sequence elements.

5.2 Simulation 5.1

Simulation 5.1 took the role of a candidate simulation to test the features of recurrence in a Chimaera network, as well as to establish any differences in the training parameters or protocols. As such, this simulation used a simple colour grammar consisting of two sequences, one 9 elements long, the other 10 elements long, with a 7-deep transitional ambiguity to test the function of recurrence.

Architecturally, the network contained two layers – an input layer, to learn the individual sequence elements themselves (in this case, colours), as well as a recurrent layer to acquire temporal sequence information. At a given epoch, the input to this recurrent layer consisted of both the activation map from the input layer (containing a pattern of activation representing a given colour element), as well as time-delayed recurrent activation from the second layer itself, representing the second layer at the previous epoch. The spatial size of each of the two layers was equal, at 32x32 nodes each (for a total of 1024 nodes per layer), such that each layer would contribute half of the resulting 2048-dimension input vector to the recurrent layer. In this way no weighting parameter was required to balance the contribution of either layer, as both layers contributed equally sized activation maps to the recurrent input vector.

Initial pilot simulations showed that the training protocol had to be modified slightly for the recurrent Chimaera network. Typically a Chimaera is trained such that first the SOM portions are trained to completion, followed by a period of association where the clusters of nodes representing a given input vector are allowed to associate and where activation is permitted to flow between nodes. While this training regime is not strictly required – associations can be made while the SOM portion of the map is also training (as described in Chapter 2) – this scheme does tend to produce a relatively low-noise map with good representations of input vectors, and little noise in the association map that may be left over from broad regions of the map associating early in SOM training.

This stage-like protocol was not compatible with training the recurrent Chimaera. The recurrent Chimaera requires that both the SOM portions and association map portions of the network are trained concurrently. Conceptually, this makes some sense, and echoes issues in training other recurrent SOM architectures (Scholtes, 1991) – as the network associates, and activation flows from one region to another, the activation map (and subsequent input vector to the recurrent layer itself) changes. In this way, the input vectors to a recurrent layer are not stable, and training in a stage-like fashion will result in a network that contains representations of normal input (from the input layer), and recurrent input from the recurrent layer – but recurrent input that is incomplete, and does not contain any associative flow. As such, any subsequent best-matching-unit judgments the network makes after flow has taken place will only be partial matches, and will lower the amount of activation the network produces (and in an extreme case, with a great deal of flow and a sufficiently specific activation function, the best-matching-unit may produce little or no activation, as a result of the additional flow present in the recurrent activation map).

For similar reasons, the training period of the recurrent network was also extended. Even without flow present in the activation map, the recurrent portion of the input vector still changes from having zero activation at the start of training, to potentially a great deal of activation by the end of training. As the recurrent activation map encompasses one half of the input vector to the recurrent layer, this means that up to one half of the input vector to the recurrent layer may continually change over the course of training (although, functionally, for a network with any representational density, much of the activation map will be inactive and near zero, with only a relatively small portion progressively increasing activation over the course of training). To encourage these changes to take place over a long period of time – allowing a greater period of adjustment and learning while the radius-of-effect of the SOM learning algorithm was still large – the recurrent layer of the network was trained for twice the duration of previous simulations.

5.2.1 *Results*

The results of the recurrent colour grammar simulation are displayed in Figure 5.1 for sequence 1 (length 10), and Figure 5.2 for sequence 2 (length 9), at epoch 96 after training. As in previous simulations, layer 1 rapidly self-organizes, producing a map that well-represents each of the input colours found across the two sequences. To facilitate analysis, the activation of layer 2 is broken down piecewise into constituent activation from the (1) input vector itself (top), (2) the input vector in concert with decaying activation from the previous epoch, (3) activation caused by flow from activation in the nodes representing the input vector flowing to their associated nodes, and (4) the sum of activation across the input vector, decay, and flow. The visualization also includes (5) refraction maps, that outline which regions of nodes contained activation on the previous epoch above the k_{s_thresh} threshold of the Chimaera network, and which will selectively associate with active nodes in the current epoch, as well as (6) a visualization of the "receptive field" of the most active unit for each of the pure input vector activation map and the flow map. This receptive field map spatially displays the data vector contained within the most active node of the respective activation map, and functions as a "check", allowing one to visualize the pattern of activation that would cause that node to become most active. Because the input vector to the recurrent layer includes activation maps from two networks (both the input layer, as well as the recurrent layer itself), the receptive field visualization too includes data from both.

Despite the input sequence set containing a 7-element-deep ambiguity, the recurrent Chimaera acquires unique representations of each element across both sequences, and successfully generates unambiguous predictions for each element in each sequence – that is to say, the recurrent layer includes separate representations for the same ambiguous element in a deeply ambiguous sequence, with each representation tied to the specific temporal context that element occurred in. Figure 5.3 shows a tagged association-map visualization, which includes labeled annotations of each representation in the recurrent layer. Interestingly, the clustering pattern of the self-organizing-map is preserved, with the global organization of the map finding identical sequence elements (in this case, colours) being clustered together, while the local organization within a region in the map shows each ambiguous sequence element containing its own unique and explicit representation that depends upon its context of presentation. This global-then-local clustering order is likely an artifact of the relative magnitude of the contribution of the input layer to the recurrent layer in the input vector to the recurrent layer. While the input layer need only represent the individual sequence elements, the recurrent layer must represent far more information – the sequence elements, as well as the contexts they appear in. As such, far more unique pieces of information exist in the recurrent layer, and as both the input and recurrent layers have the same number of nodes, the representational density of the recurrent layer will be greater than that of the input layer, and proportionally less of the recurrent layer will be active at a given epoch. As such, input from the input layer, representing the sequence element itself (in this case, the colour), will have proportionally more nodes active to a greater degree in the input vector than the recurrent layer, and exert more influence on the best-matching-unit function of the SOM learning algorithm that ultimately determines the topographic organization of the representations.

5.3 Simulation 5.2

Where Simulation 5.1 examined the ability of the recurrent Chimaera network to acquire an input set that contained a deep ambiguity, but was otherwise relatively easy in terms of the total number of sequences (2), total number of ambiguities (1), and in containing sequence elements that were relatively different from one another, Simulation 5.2 switched focus to examine the network's ability to acquire a commonly used and linguistic-grammar input set that, while containing only three sequences, also contains a number of transitional ambiguities and frequent repetition of the sequence elements.

The input sequence set used in simulation 5.2 is shown in Table 5.1. The set is a common quantitative benchmark for linguistic grammar learning in neural networks, and was originally used by Van der Velde et al. (2004) as a measure of systematicity in connectionist systems. The set includes abstractions of a simple sentence structure (N – V – N), as well as variations that include a right-branching clause (N – V – N – Who – V – N), and centre-embedded clause (N – Who – N – V – V – N). As in previous simulations, each element in the input set was mapped to a unique 3-dimensional input vector representing a colour to facilitate visualization, and these colours were chosen to be distant and dissimilar enough from each other as to not produce overlapping patterns of activation.

The network architecture of Simulation 5.1 was altered slightly to reduce noise and promote the generation of unique patterns of activation for each sequence element in the highest layers of the network. The network was increased from a 2-layer network to a 3-layer network, with an input layer, recurrent layer, and output layer, similar to the architecture of the simple recurrent network. The intra-layer recurrence of the Chimaera, which allows the association and flow of activation within a layer, as well as the generation of a single-layer prediction mechanism, was included for the output layer (in order to generate a prediction for the next element in the sequence), but was otherwise not included for input or recurrence layers. This was both to facilitate speed of simulation (the intra-layer association mechanics are computationally intensive), as well as because this intra-layer recurrence did not appear to impart any processing advantage.

Visualizations for the network at epoch 160, after learning has completed, are present in Figure 5.4 (simple), Figure 5.5 (right-branching), and Figure 5.6 (centre-embedded). The complexity of the input data set (in terms of the number and depth of transitional ambiguities), as well as the complexity of the network (in terms of the number of states present across the recurrent and output layers of the network) had reached sufficient complexity as to make functional analysis by hand-inspection and tagging daunting. As such, here we transition from analyzing the network dynamics and representational function by a process of hand-inspection to a much more overarching quantitative performance metric based on grammatical prediction error. For the purposes of comparison, we also transition from using hand-crafted input sets to the benchmark dataset of Van der Velde et al. (2004).

5.3.1 Grammatical Prediction Error and Analysis Metrics

Network performance in the next-word-prediction task is often evaluated by measuring the grammatical prediction error (GPE, Van der Velde et al., 2005), or the probability of the network to generate an erroneous and ungrammatical prediction for a transition from a given part-of-speech to the next part-of-speech in a parse. Because the Chimaera does not include a single output unit for a given part of speech, but rather the possibility for a number of clusters of units, the GPE measure had to be adapted to fit the output of the network.

To obtain a conservative measure of network performance, an analysis method of automatically tagging the involvement of each node in the activation map of the output layer was developed, and then used to quantify the predictive activation present in the flow maps for each sequence element. First, the activation map of the output network was automatically tagged at the end of training, such that for a given element in a sequence parse (for example, NOUN), any nodes found active in the activation map of the output layer (above an activation threshold) would be categorized as representing that sequence element. This process was repeated for each element across all sequence sets (allowing for the possibility that a given node may be active for more than one element), and a final tagged map was obtained.

To obtain a measure of GPE, each sequence was serially presented to the network as in training, where the flow map of the output network – representing the predictive activation for possible transitions to the next part-of-speech – was analyzed using the activation map key tagged for part-of-speech, as above. For each output node in the flow map, if that node was above a low noise threshold, the node would be categorized as generating a prediction for any tag(s) associated with that node. The proportion of nodes generating predictions for each tag is then calculated, generating a probability distribution for each part-of-speech to be predicted as the next transition for a given location in the parse. Note that this method is particularly conservative, and does not take the magnitude of the flow of activation to a given node into account, beyond a low noise threshold. To further increase the sensitivity of this measure, as well as mediate any effects of association magnitude across the data set (which can be caused by relative presentation frequency, as observed in Chapter 3, Figure 3.3), each flow map was normalized to between zero and one before analysis. As such, if we choose our thresholds appropriately, we will obtain an extremely sensitive, worst-case measure of accuracy.

The results of the GPE analysis are displayed in Figure 5.7, where the mean GPE across each sentence type is displayed in Table 5.2. For these analyses, the activation threshold for being tagged with a given part-of-speech was set at being active greater than 0.50, while the noise threshold in the flow map was set to 0.10 (after normalization). Compared to the k_{s_thresh} refraction threshold of 0.80, these parameters ensure that we will be analyzing not only the nodes that are specifically involved in flow to other regions, but also the general cluster of lesser-active nodes surrounding the centrally active cluster. The tagging threshold of 0.50 was selected to be low enough to facilitate the large central cluster of each representation to be tagged, while carefully avoiding tagging the less active regions of the perimeter bordering adjacent representations.

The mean GPE analysis shows that the network generates erroneous grammatical predictions, on average, about 9.4% of the time. Examining the transition prediction probabilities for each element across all sequences in Figure 5.7, we note that the network has generally good performance in predicting grammatical transitions, with the probability of the transition approximately representing the frequency of presentation for a given transition in the training set (for example, the fourth element of simple and right-branching parses can transition either to an end (in the case of the simple parse), or a "who" element (in the case of the right-branching parse) about 50% of the time, reflecting that equal numbers of these parses were seen during network training). Erroneous predictions (marked with asterisks) appear to be localized to a single incorrect prediction for each transition, reflecting mild levels of self-association (visible on the flow maps of Figures 5.4 through 5.6), and when these self-association predictions are removed, the network is nearly 100% accurate.

5.4 Discussion

The network generates good grammatical prediction performance, with a mean accuracy of 90.6% . Where incorrect predictions due to self-association are removed, the performance accuracy of the network reaches nearly 100%. Compared to the supervised models of Frank (2006) and the unsupervised model of Farkas and Crocker (2008), the network exhibits approximately 9% poorer performance on training sets than either the SRN or Echo-State network model of Frank (2006), or the RecSOMsard model of Farkas and Crocker (2008), and comparable performance when self-association is removed – although these networks are performing a much more difficult task.

While the grammar prediction models of Frank (2006) and Farkas and Crocker (2008) are tested on familiar training sets, the models themselves are designed to attempt to abstract grammatical information from these training sets, and apply this grammatical knowledge to the successful prediction of sentences composed of familiar words in novel sentences – a measure of grammatical systematicity (Hadley, 1994). The recurrent Chimaera model presented here has made significant progress over earlier Chimaera models in that it is able to successfully acquire a benchmark test grammar that includes a variety of deep transitional ambiguities amongst a small vocabulary of sequence elements with near-perfect accuracy, but the model is still performing a very different task than those charged with acquiring grammatical systematicity. A critical demonstration of grammatical systematicity is to acquire grammatical knowledge from the language stream itself, at the level of the sentence and sequences of words, rather than at a level that supplies pre-parsed grammatical input. While the recurrent Chimaera model has demonstrated that it can acquire a grammar, it is currently unable to acquire this grammatical knowledge from a stream of words, or to operate at the lexical level, and as such requires further extensions to compete on the level of grammatical systematicity.

Sentence Type	Sentence Structure					
Simple	N	V	N			
Right	N	V	N	Who	V	N
Centre	N	Who	N	V	V	N

Table 5.1: The van der Velde et al. (2004) grammar

	Performance (GPE)		
	Simple	Right	Centre
2S6	0.092712 (0.016238)	0.105369 (0.020396)	0.082343(0.016733)

Table 5.2: Mean grammatical prediction error (GPE) across simple, right-embedded, and centre-embedded sentence structures in Simulation 5.2.

Figure Captions:

Figure 5.1: The results of Simulation 5.1 for the 7-back colour grammar for sequence 1 (length 10). Layer 1 acquires representations of the individual sequence elements, where Layer 2 acquires representations of the transitions between sequence elements, and generates valid temporal sequence predictions.

Figure 5.1: The results of Simulation 5.1 for the 7-back colour grammar for sequence 2 (length 9). Layer 1 acquires representations of the individual sequence elements, where Layer 2 acquires representations of the transitions between sequence elements, and generates valid temporal sequence predictions.

Figure 5.3: A tagged association map for Simulation 5.1 Layer 2. Topographic clustering shows a global preference for sequence element, with local clustering based on temporal context. Values in parentheses represent which sequence a given temporal representation is part of.

Figure 5.4: The results of Simulation 5.2 for the simple sentence structure. The input layer (layer 1) represents individual parts of speech, where the output layer (layer 3) contains distinct representations for each transition between sequence elements in the van der Velde et al. grammar.

Figure 5.5: The results of Simulation 5.2 for the right-branching sentence structure. The input layer (layer 1) represents individual parts of speech, where the output layer (layer 3) contains distinct representations for each transition between sequence elements in the van der Velde et al. grammar.

Figure 5.6: The results of Simulation 5.2 for the centre-embedded sentence structure. The input layer (layer 1) represents individual parts of speech, where the output layer (layer 3) contains distinct representations for each transition between sequence elements in the van der Velde et al. grammar.

Figure 5.7: Transitional prediction values across each possible transition in the van der Velde et al. grammar. The network displays generally good performance, with mild levels of self-association visible. Ungrammatical transitions are marked with an asterisk.

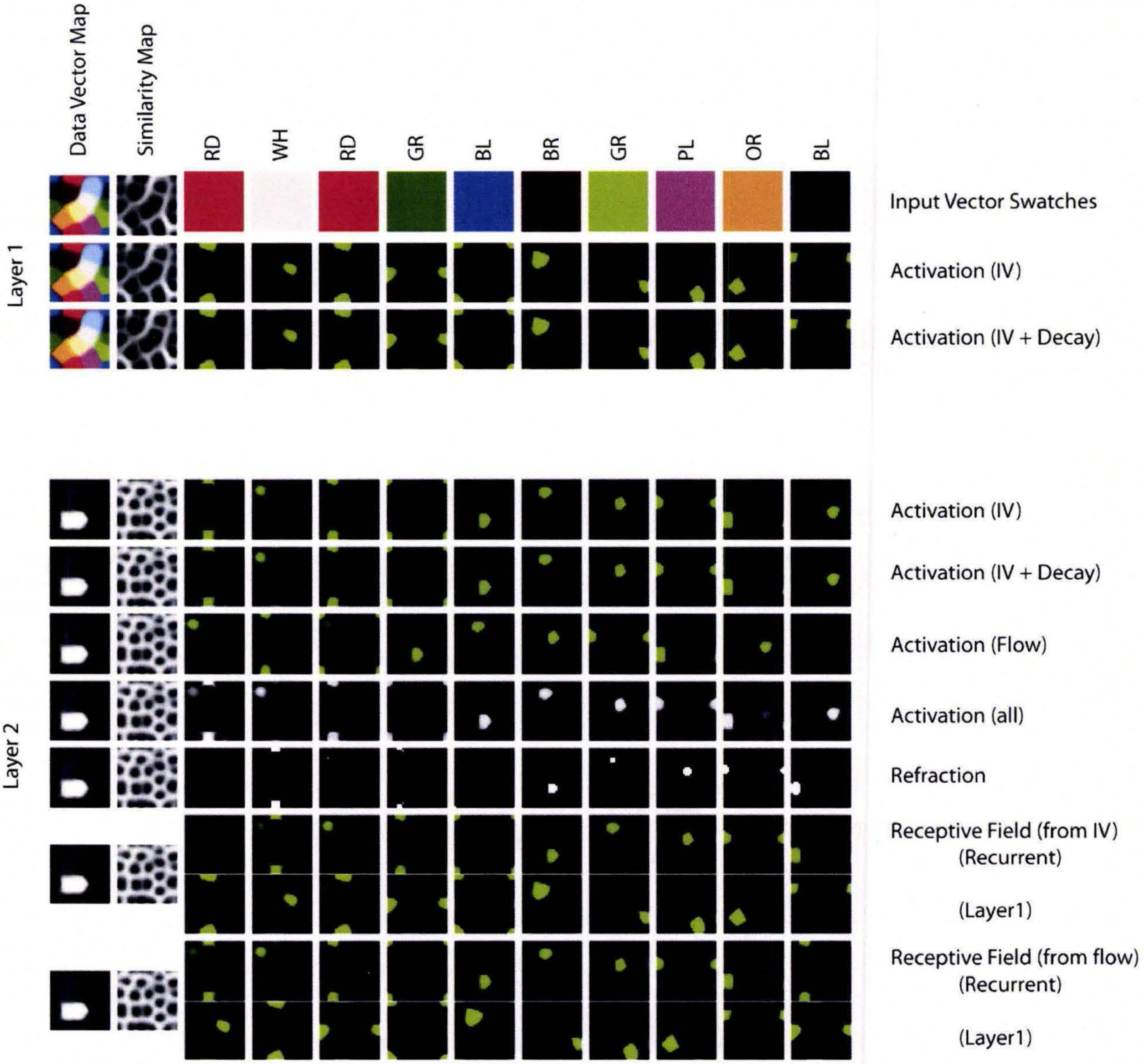

Data Vector Map
Similarity Map
RD
WH
RD
GR
BL
BR
GR
PL
OR
BL
Layer 1
Input Vector Swatches
Activation (IV)
Activation (IV + Decay)
Layer 2
Activation (IV)
Activation (IV + Decay)
Activation (Flow)
Activation (all)
Refraction
Receptive Field (from IV)
(Recurrent)
(Layer1)
Receptive Field (from flow)
(Recurrent)
(Layer1)

Figure 5.1

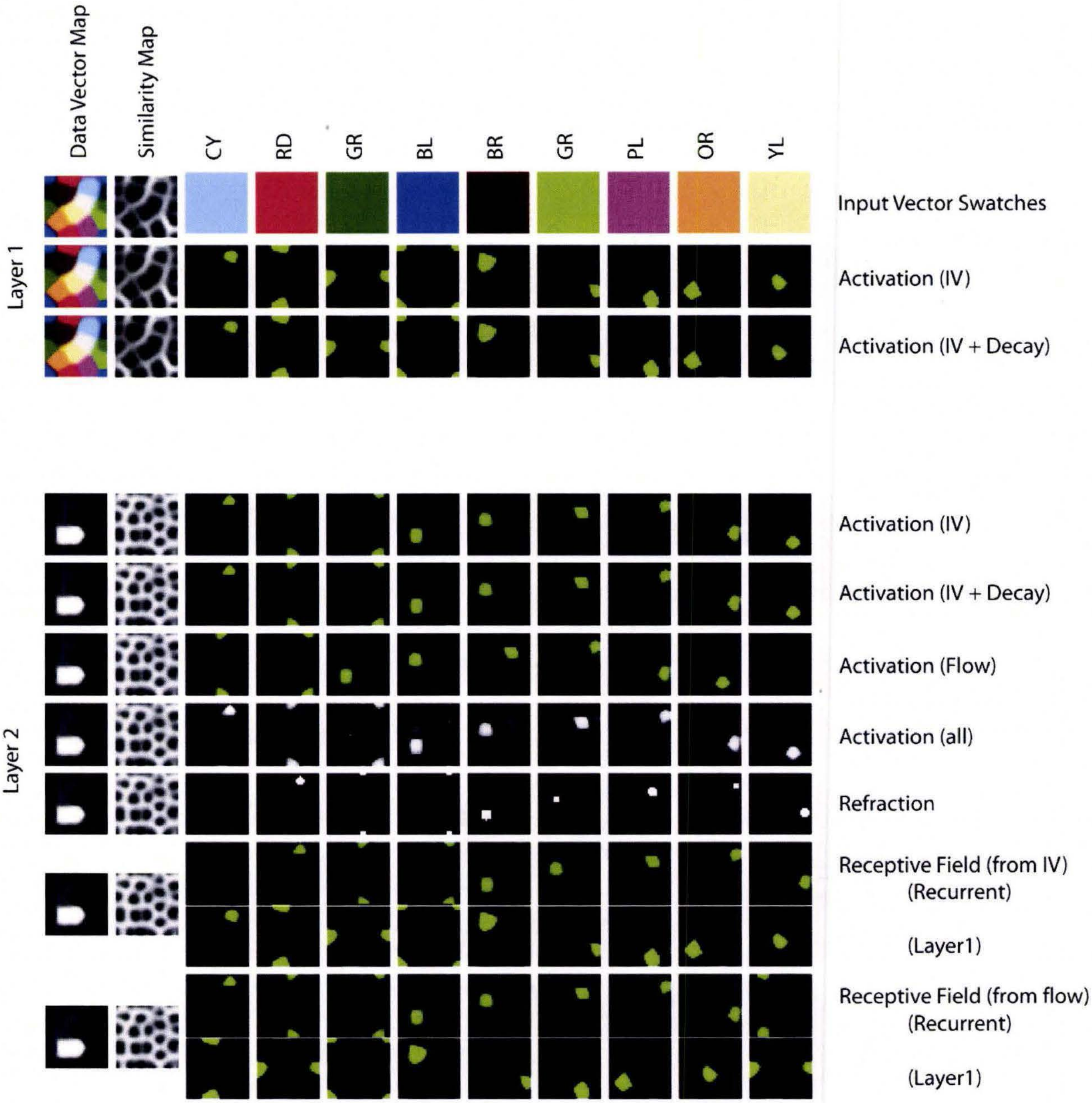

Data Vector Map
Similarity Map
CY
RD
GR
BL
BR
GR
PL
OR
YL
Input Vector Swatches
Layer 1
Activation (IV)
Activation (IV + Decay)
Layer 2
Activation (IV)
Activation (IV + Decay)
Activation (Flow)
Activation (all)
Refraction
Receptive Field (from IV)
(Recurrent)
(Layer1)
Receptive Field (from flow)
(Recurrent)
(Layer1)

Figure 5.2

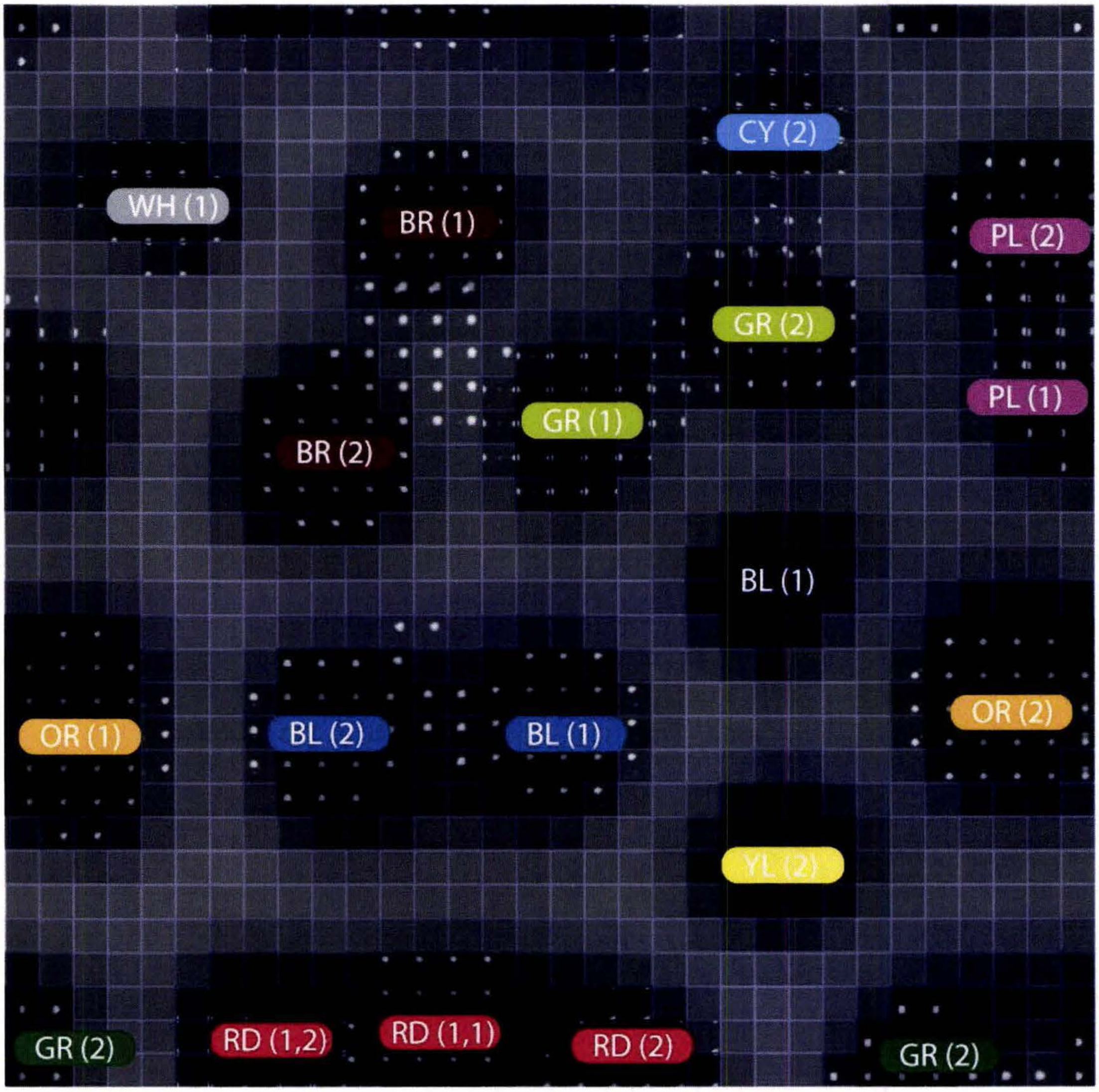

CY (2)
WH (1)
BR (1)
PL (2)
GR (2)
GR (1)
BR (2)
PL (1)
BL (1)
OR (1)
BL (2)
BL (1)
OR (2)
YL (2)
GR (2)
RD (1,2)
RD (1,1)
RD (2)
GR (2)

Figure 5.3

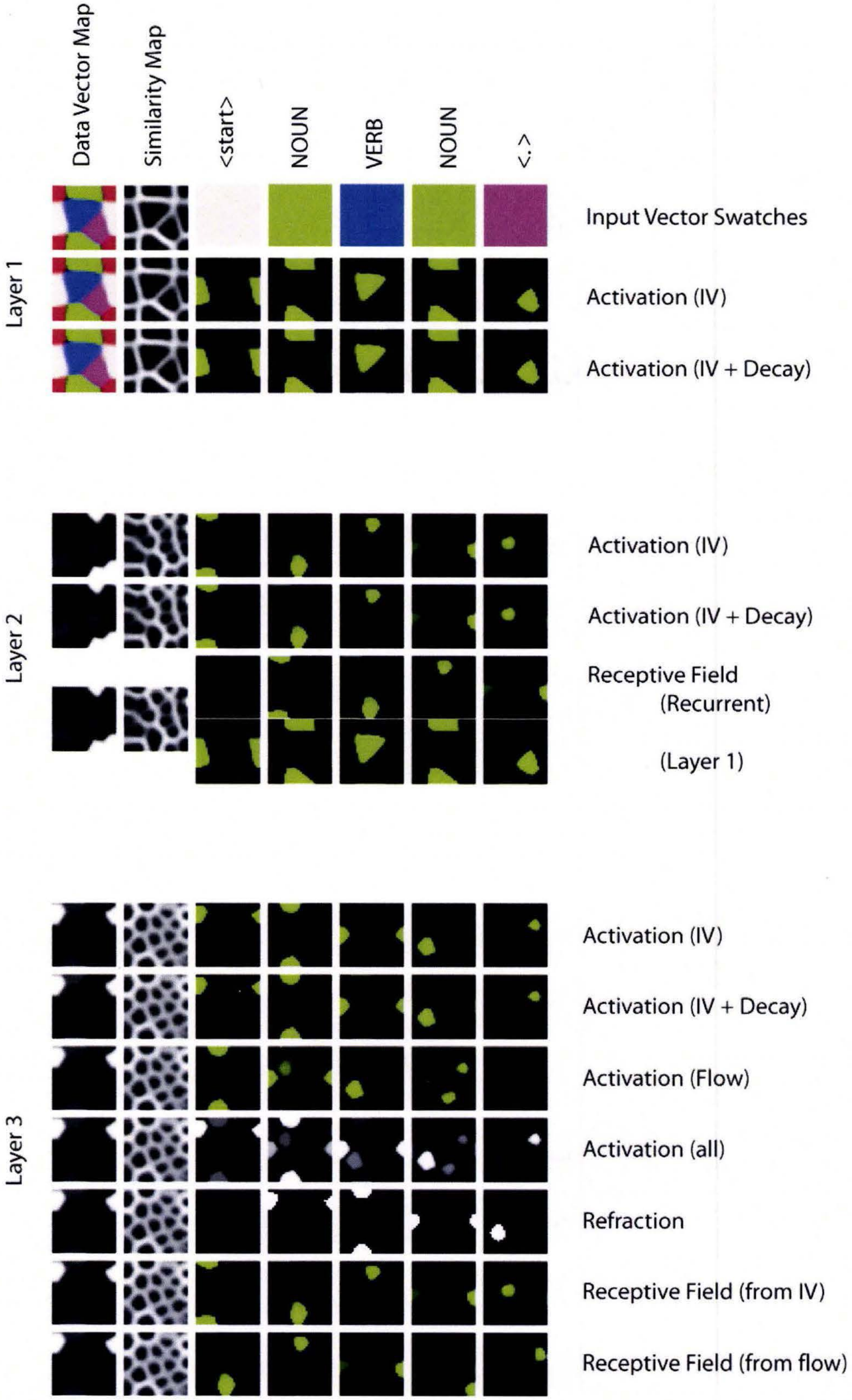

Data Vector Map
Similarity Map
<start>
NOUN
VERB
NOUN
<.>
Layer 1
Input Vector Swatches
Activation (IV)
Activation (IV + Decay)
Layer 2
Activation (IV)
Activation (IV + Decay)
Receptive Field
(Recurrent)
(Layer 1)
Layer 3
Activation (IV)
Activation (IV + Decay)
Activation (Flow)
Activation (all)
Refraction
Receptive Field (from IV)
Receptive Field (from flow)

Figure 5.4

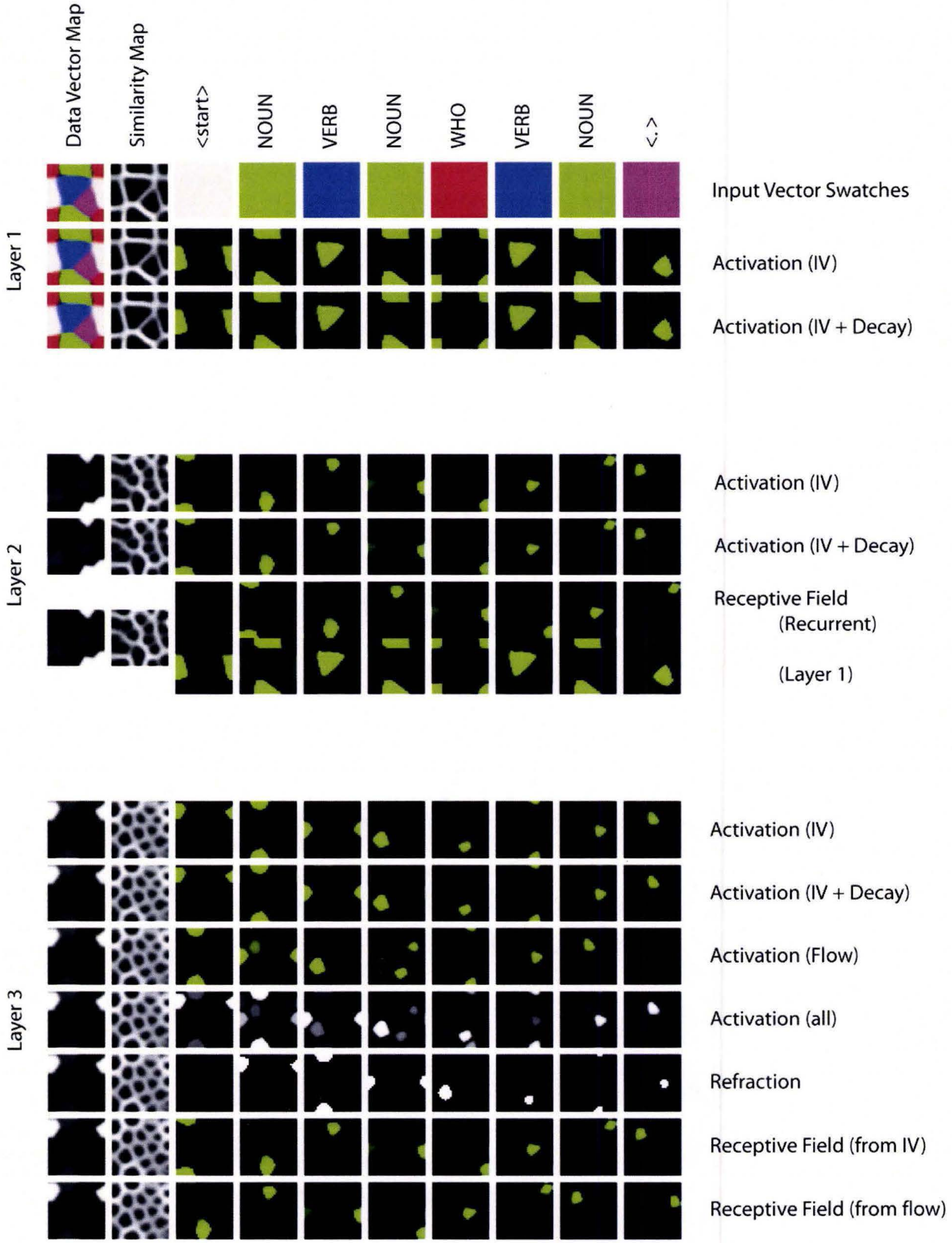

Data Vector Map
Similarity Map
<start>
NOUN
VERB
NOUN
WHO
VERB
NOUN
<.>
Layer 1
Input Vector Swatches
Activation (IV)
Activation (IV + Decay)
Layer 2
Activation (IV)
Activation (IV + Decay)
Receptive Field
(Recurrent)
(Layer 1)
Layer 3
Activation (IV)
Activation (IV + Decay)
Activation (Flow)
Activation (all)
Refraction
Receptive Field (from IV)
Receptive Field (from flow)

Figure 5.5

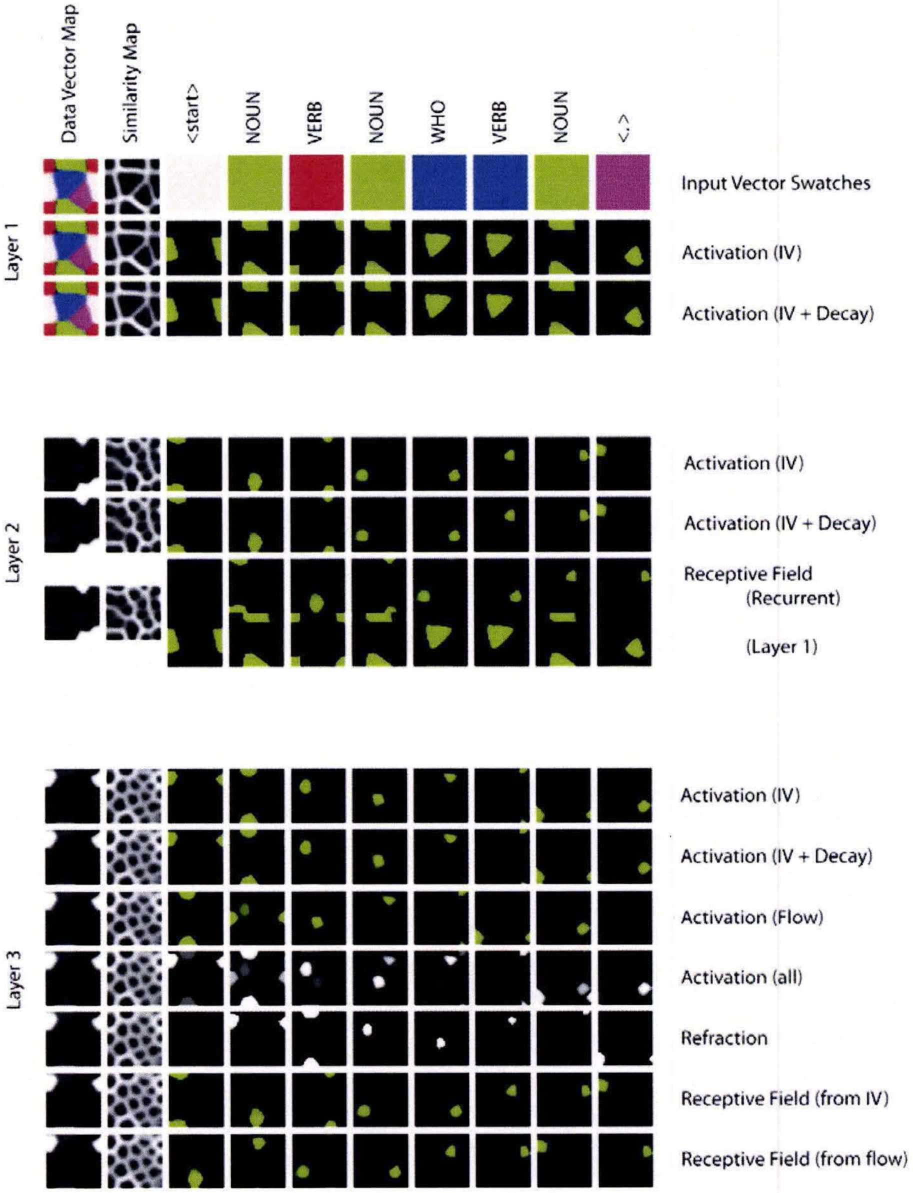

Data Vector Map
Similarity Map
<start>
NOUN
VERB
NOUN
WHO
VERB
NOUN
<.>
Layer 1
Input Vector Swatches
Activation (IV)
Activation (IV + Decay)
Layer 2
Activation (IV)
Activation (IV + Decay)
Receptive Field
(Recurrent)
(Layer 1)
Layer 3
Activation (IV)
Activation (IV + Decay)
Activation (Flow)
Activation (all)
Refraction
Receptive Field (from IV)
Receptive Field (from flow)

Figure 5.6

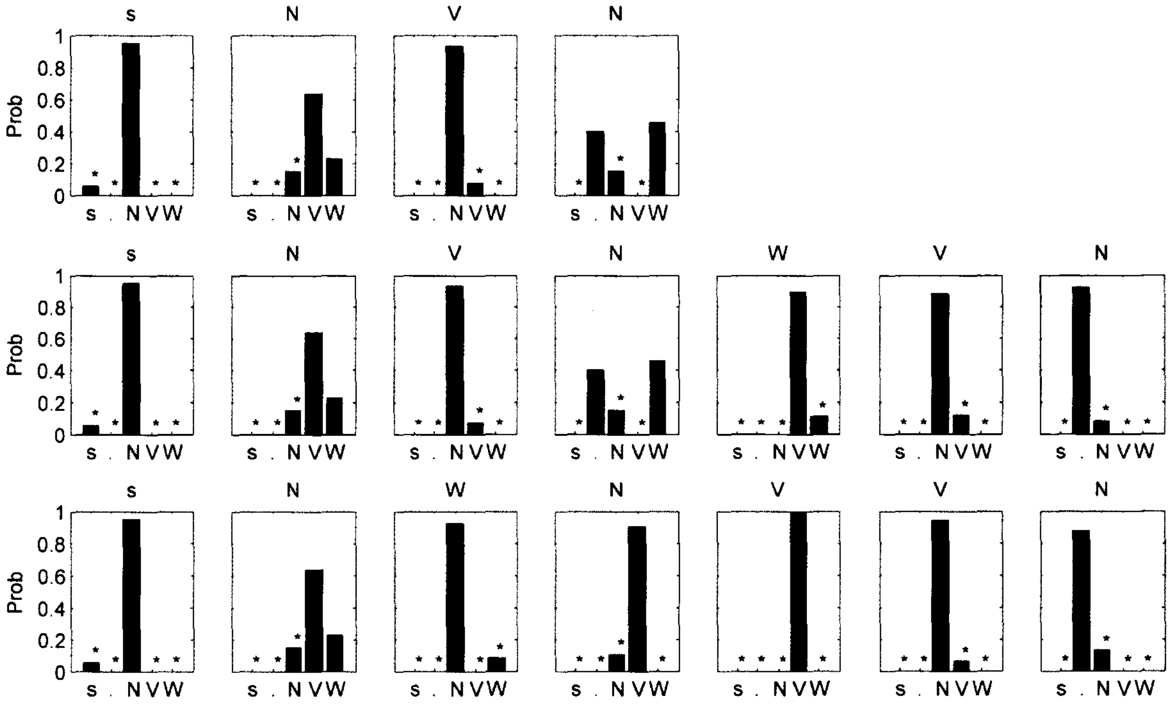


Figure 5.7

Chapter 6

Co-occurrence-based semantic distillation of grammatical class using developmentally-plausible sensorimotor conceptual input

6.1 Introduction

Taking a step back from language modeling, the problem of abstracting the deep underlying structure of some arbitrary and diverse input stream in an effort to uncover a set of rules and principles that have been used to generate a potentially overwhelming set of examples seems an incredible and potentially intractable task. This seems particularly true in light of artificial grammar learning work, that generally finds adult humans are not adept at learning even mildly complex grammars, especially as the depth of the ambiguity in those grammars increases (e.g. van den Bos, Poletiuk, 2008).

Infants, on the other hand, have remarkable success with this problem. In the domain of language, young infants rapidly begin to acquire speech sounds, beginning to parse words from streams of phonemes by as early as 4.5 months (Mandel, Jusczyk, and Pisoni, 1995). This is followed by acquiring a handful of their first words, before a sudden vocabulary explosion that rapidly increases the infant's lexical diversity. By the age of five most children are fluent speakers of their native language, having distilled a general knowledge of the combinatorial grammar that describes the structure of language across all levels of representation – whether it be combinations of phonemes that create valid morphemes, or combinations of words that create grammatical sentences. While adults struggle with even simple artificial grammar learning, infants are solving a general and much more diverse version of this problem, three orders again deep! From the level of speech sounds, infants somehow arrive at a useful knowledge of language comprehension and the underlying lexical and grammatical knowledge that task encompasses in only a few short years. How might they accomplish this?

The task of the infant is mediated, at least in part, by virtue of beginning small, initially learning the aspects of a language piece by piece, word by word (Brown, 1973). They also benefit from having access to a rich pre-linguistic conceptual system (Mandler, 2004) ready to begin to accept stable linguistic labels supplied by a caregiver for their early conceptual mental representations. Here, we examine how access to pre-linguistic conceptual representations can assist in acquiring a knowledge of grammatical category, and ultimately enable systematicity in connectionist models of grammar acquisition.

6.1.1 *Transitioning from abstract input vectors to grounded sensorimotor representations*

From the perspective of connectionist modeling, distilling grammatical knowledge from a corpus of sentences is a task traditionally abstracted away to entirely symbolic levels, where the lexical representation of a given word is functionally represented as a unique number – much as one might expect were they learning a grammar solely from an ungrounded text – devoid of content, semantics, or any other knowledge at the level of the word. It is often argued that this difficult task is a fairly pure generalization of the task of grammar learning, and that a successful demonstration of grammar acquisition and grammatical systematicity from such a symbolic input set would be a particularly powerful demonstration of a neural network model's ability to generalize. While some models have approached grammar learning this way (e.g. Farkas and Crocker, 2008; Frank, 2006; Van der Velde, 2004), attempting to distill general underlying structure from no more than a large corpus of symbolic instances, others (e.g. Howell et al., 2005) have approached the problem from the perspective of symbol grounding (Harnad, 1990) and embodied cognition (e.g. Clark, 2008). From this perspective, a young infant learning language is greeted not with simply a number (or a symbol), but rather a symbolic label that can be paired with an already rich pre-linguistic conceptual representation that shares a great deal of similarities between other conceptual representations. It is these semantic commonalities and distinctions that allow the infant to tease initial broad category knowledge from their linguistic stream, and enable their initial acquisition of grammatical knowledge – a perspective known as semantic bootstrapping (Pinker, 1984).

One of the difficulties in transitioning from abstract symbolic input vectors to those that have some grounded character is the task of generating such an input set. In their effort to determine the effects of grounded representations on grammar learning in simple recurrent networks, Howell et al. (2004) constructed a database of developmentally-plausible sensorimotor concept vectors containing rich feature representations of 352 nouns and 89 verbs across 97 noun-feature and 84 verb-feature dimensions. Howell at al. took great care to ensure these feature dimensions were human-generated, non-artificial, and developmentally plausible. Beginning with the thousands of human-generated semantic features of McRae et al. (1997), Howell et al. extracted noun features describing explicitly perceptual knowledge, then had these feature dimensions independently rated for the plausibility that an 8- to 28-month infant would be sensitive to them. Similarly, verb features were generated by undergraduate participants, then later teased apart to arrive at 84 verb feature dimensions that were largely kinesthetically-themed (such as "requires head motion"), or described changes of state (as in "decreases hunger"). A total of 352 concrete nouns were selected from the MacArthur Communicative Development Inventory (MCDI, Fenson et al., 2000) and rated across the 97 feature dimensions by undergraduate participants, while 89 early and/or prototypical verbs from the MCDI as well as Goldberg (1999) were similarly rated. Participants were asked to ascribe the likelihood of a given word to contain a given feature dimension, and assign this likelihood a value between 1 and 10. These values were averaged, then later normalized to arrive at a scalar probability for each feature across all words. The cluster analysis of Howell et. al showed this dataset contains very good overall categorical agreement within nouns and verbs at 88% and 70%, respectively, where chance performance was 9.1% for nouns, and 11.1% for verbs.

6.1.2 Co-occurrence based activation function

From the perspective of a self-organizing neural network, how might one acquire broad categorical knowledge from semantic representations? One possible method comes from the notion of semantic similarity – already a low-level property of the activation function of a Chimaera network. At a given epoch, the Chimaera network generates an activation map that contains positive values of activation for nodes whose data vector representations are within a defined Euclidian distance from the input vector. By simply increasing the range of this Euclidian distance, the activation function might be made to produce positive values of activation for all data vectors that are similar to a given input vector – and using this broad signal as input, we might have a measure of category. But first, we must consider how we define similarity. Consider the following set of input vectors:

Vector Number	Feature Dimension	
	Loudness	Colourfulness
1	0.0	0.9
2	0.1	0.0
3	0.0	0.1
...		

Table 6.1: Sample input vectors representing semantic co-occurrence

Were a Chimaera network fully trained on these input vectors, then presented with the third input vector and asked to activate all similar representations (defined as all representations out to some Euclidian distance), the activation function would much sooner activate vector 2 ($||v_3-v_2|| = 0.14$)

than vector 1 ($||v_3-v_1|| = 0.8$) despite vectors 2 and 3 having no commonalities aside from the relative intensities of their respective features – but sharing no features in common. One might argue that semantically, vectors 1 and 3 are much more similar than vectors 2 and 3 because they share the same features, with one vector simply being a less intense version of the other – the difference is of *degree*, and not of *kind*. As such, Euclidian distance seems a poor measure of semantic similarity, and cross-category activation is almost certain to occur.

We might define a metric for the semantic similarity of concepts expressed as vectors as follows: (1) two vectors are increasingly similar as the number of features they have in common increases, and (2) two vectors are increasingly similar the closer their values match on a given feature dimension. In essence, this definition of similarity is based on semantic feature co-occurrence, rather than mathematically on Euclidian distance.

Formally, the semantic co-occurrence feature metric ***m*** is defined as:

$$m(i,t) = \vec{x}(t) \circ \vec{w}_i(t) \tag{1}$$

which is the entrywise product of the input feature vector ***x(t)*** and the SOM weight vector ***w(t)*** for a given node ***i*** . The Euclidian activation function ***y_i(t)*** is then replaced by the semantic co-occurrence activation function:

$$y_i(t) = \frac{1}{\tau}\left(\frac{\sum_{a=0}^{b} m'_a(i,t)}{\sum_{a=0}^{b} m''_a(i,t)} - 1 \right) + 1 \tag{2}$$

which is an adaptive scoring metric based on the presence of non-zero (and greater than m_{thresh}) feature values on each dimension $a \in (1,2,...,b)$ of feature vectors ***x(t)*** and ***w_i(t)***. m', m'' function as score parameters marking the current (m') and maximum possible (m'') scores on each dimension a:

$$\begin{aligned}
&m'_a(i,t) = \arg\max\left(\{0, w_{ia}(t) - |w_{ia}(t) - x_a(t)|\}\right),\\
&m''_a(i,t) = w_{ia}(t),\\
&\quad for\ w_{ia}(t) > m_{thresh} \vee x_a(t) > m_{thresh}\\
&\\
&m'_a(i,t) = 0,\\
&m''_a(i,t) = 0,\\
&\quad for\ w_{ia}(t) \le m_{thresh} \wedge x_a(t) \le m_{thresh}
\end{aligned} \tag{3}$$

and τ provides a scaling parameter to determine what degree of co-occurrence will generate a positive activation value (where values close to 0 signify vectors must be very similar to produce positive activation values, and values close to 1 signify that the vectors need only generally share features to produce positive activation values).

The semantic co-occurrence activation function ***$y_i(t)$*** is defined on (0,1), where negative values are clipped to 0, and the greater the value, the greater the semantic feature co-occurrence between two vectors. Note that in the case where two vectors share none of the same features, the resulting similarity value based on co-occurrence will be 0. Intuitively, this makes a bit of sense – while we could define a metric that might fall back to another measure of distance should co-occurrence not yield a match, from the perspective of semantic similarity it would not be meaningful to ascribe (for example) whether vector 2 from Table 6.1 was closer to a vector that was soft, versus one that was tangy. In this way the semantic feature co-occurrence metric is somewhat non-invertable – in cases where it is unable to detect the similarity between vectors, it is also unable to provide a measure of dissimilarity.

6.2 Simulation 6.1

The Chimaera activation function and best-matching-node search (for the SOM learning rule) were modified such that they each made use of the semantic co-occurrence metric defined above, rather than a Euclidian metric. Specifically, the activation of a node given the input vector ***x(t)*** is defined in (2), where the best-matching-node search function was similarly modified such that the search aimed to maximize the value of the semantic co-occurrence similarity metric between a given node and the input vector. The index of the best matching unit *k* is defined as:

$$k(t) = \arg\max_{i \in (1,2,\ldots,N)} s_i(t) \qquad (4)$$

The architecture of the network in this simulation consists of two layers, where the input layer will acquire and represent 50 sensorimotor nouns and verbs, while layer 2 is designed to represent at the part-of-speech level, and acquire representations of the part-of-speech of the concept presented to the input layer. Both layers make use of the semantic co-occurrence activation function (2). Where this co-occurrence activation function serves to activate semantically similar sensorimotor concepts in the input layer, when applied to an input vector that represents the processing context of a subordinate layer, the activation function will provide positive activation for spatial areas that tend to have non-zero activation values that co-occur in the processing context of the input layer. In this way, this two layer system functions as a semantic "low-pass" filter, producing broad categorization based on semantic similarity as defined by the semantic co-occurrence metric. Appendix A includes the network parameters describing this simulation.

6.2.1 Input Set

The input set consisted of 50 randomly selected nouns and verbs (25 of each), such as MOMMY, PLAY, and OWL, from the developmentally-plausible sensorimotor concept data set of Howell et al. (2005). Of the 181 sensorimotor feature dimensions in this data set, 97 are entirely ascribed to nouns, while the remaining 84 describe verbs, and as such these sensorimotor-tagged nouns and verbs do not contain any co-occurring features between grammatical category.

6.2.2 Results

The results of Simulation 6.1 are displayed in Figure 6.1. After 64 epochs, the network has fully trained, and contains distinct sensorimotor representations for each of the 50 nouns and verbs in the input set. The activation maps of the input layer show broad patterns of activation for both nouns and verbs. While the spatial regions involved in activating a given noun or verb do not overlap across grammatical category, and as such the collection of all nouns or all verbs share the same nodes, the activation values across these groups of nodes are unique for each sensorimotor concept. As such, while the spatial pattern is similar, the pattern of activation in the network for a given node is unique to each word, though similar sensorimotor concepts will produce similar but unique patterns of activation.

By layer two, the co-occurrence activation function has distilled the grammatical category of each input vector based on the spatial co-occurrence of the sensorimotor noun and verb representations of the input layer. Where the input layer showed a great deal of variation in the activation pattern for a given word, the activation pattern in layer two shows much less variation, and is virtually identical within grammatical category. The spatial clustering of nouns and verbs in layer 1, as well as the clustering of grammatical category in layer 2, is displayed in Figure 6.2.

6.3 Simulation 6.2

Where Simulation 6.1 successfully demonstrated the abstraction of grammatical category based on semantic feature co-occurrence, Simulation 6.2 aimed to incorporate this function into a broader model able to acquire both grammatical category information from sensorimotor word representations (as in Simulation 5.1), as well as a broader knowledge of grammatical combinations of parts of speech (as in Simulation 5.2). As such, this simulation aimed to take sensorimotor representations of words as input (although one could just as easily take lexical representations of words that are then transduced into sensorimotor representations by a 2-layer feed-forward network – the distinction is trivial), and to acquire a knowledge of both the part of speech of a given sensorimotor concept, as well as a knowledge of grammatical combinations of words such that a grammatical prediction could be successfully generated for the next-word prediction task.

The architecture of the network was optimized to promote the best possible grammatical prediction performance. The network layout was functionally the two-layer grammatical category distillation architecture from Simulation 5.1, which served as input to the recurrent grammar sequence learning network of Simulation 5.2 . While, strictly speaking, the input layer of Simulation 5.2 should be redundant and not required to serve input to the recurrent layer, this input layer was included, and transitioned from a Euclidian activation function to a semantic co-occurrence function to promote even further generalization from the grammatical category network (and increase the intensity of the "semantic low-pass filter" of Simulation 6.1). Pilot simulations showed that the recurrent grammar sequence learning network was not amenable to the semantic co-occurrence metric, and as such the remainder of this network used Euclidian-based activation and best-matching-node metrics, as in Simulation 5.2. The details of this simulation are included in Appendix B.

6.3.1 Input Set

The input set was divided into a training set of 9 sentences, and a test set of 30 sentences. The structure of these sentences took the form of equal numbers of simple (N – V – N), right-branching (N – V – N – Who – V – N), and center-embedded sentence structures (N – Who – N – V – V – N), as in the case of Simulation 5.2, as well as Frank (2006) and Farkas and Crocker (2008). The 9 training sentences contained a total of 24 unique nouns and 15 unique verbs, where the test sentences were randomly generated and contained 80 instances of nouns and 50 instances of verbs from the remaining 402 sensorimotor concepts. That is, the test and training sets contained *zero* overlapping nouns or verbs. Examples from both the training and test sets are found in Table 6.2. To ascribe vector representations to the start marker, end marker, and "who" clause marker, arbitrary independent input vectors were created that did not share any feature dimensions with either the sensorimotor nouns and verbs, or each other.

6.3.2 Results

The network's grammatical prediction error (GPE) was analyzed using the analysis metric defined in Chapter 5. Mean GPE is displayed in Table 6.3, where GPE for each transition across each sentence type (simple, right-branching, and centre-embedded) is displayed in Figures 6.6 and 6.7 (where analyses are grouped across training and test sets, as there was functionally no difference in performance between the two sets). The results show impressive performance across all transitions, where mean GPE across all sentence types was 1.3% (32x32 simulation) and 1.7% (20x20 simulation). Figures 6.6 and 6.7 show very minor contributions of self-association to error, with otherwise excellent performance predicting each possible grammatical transition.

Figures 6.3 and 6.4 show the activation patterns across the network for a familiar center-embedded sentence found in the training set, and a novel randomly generated centre-embedded test sentence, respectively. Across the grammatical sequence learning network of layers 3 through 5, virtually no difference in the activation patterns between familiar and novel sentences is present, suggesting the grammatical category distillation mechanism of layers 1 and 2 is operating effectively, and delivering broad part-of-speech information to the superordinate layers. Figure 6.5 shows a tagged version of the association map of layer 5, the output layer of the network, where the representational states for a given part-of-speech, at a given temporal context, are displayed. The global clustering shows a preference for sentence structure, where the constituent parts-of-speech for simple, right-branching, and centre-embedded sentences are clustered together. Overall, this pattern of qualitative representational clustering and quantitative grammatical prediction performance is consistent with a network that has distilled both a broad knowledge of grammatical category and an excellent knowledge of the grammatical sequences these parts of speech can combine into. The network is then generally able to use this knowledge to generate predictions for possible grammatical transitions from a given word in a sentence with near perfect accuracy.

Sentence Type	Training Sentences (9 of 9 shown)					
Simple	MOMMY	STOP	SHOWER			
	UNCLE	DANCE	PORCH			
	CAT	SIT	BENCH			
Right-branching	FROG	WHO	GIRL	PAINT	HUG	SPRINKLER
	PEOPLE	WHO	NURSE	DRIVE	TOUCH	HELICOPTER
	BUNNY	WHO	MONKEY	SMELL	RUN	ANT
Centre-embedded	TEACHER	PRETEND	BUTTERYFLY	WHO	SEE	BUG
	GRANDPA	WRITE	DOCTOR	WHO	READ	BOOK
	AUNT	SHOW	ELEPHANT	WHO	BLOW	NOSE

Sentence Type	Test Sentences (9 of 30 shown)					
Simple	SNOWSUIT	COOK	DOLL			
	TURTLE	WATCH	TELEPHONE			
	GLASSES	SLIDE	TRASH			
Right-branching	TISSUE	WHO	MOUSE	CHASE	SLIDE	SLED
	SPOON	WHO	OWL	CHASE	CUT	VANILLA
	DUCK	WHO	BOWL	OPEN	TALK	FRENCHFRIES
Centre-embedded	BROOM	WAKE	MILK	WHO	WATCH	LIVINGROOM
	PARTY	LOVE	SHORTS	WHO	TICKLE	HAMMER
	JELLY	JUMP	TREE	WHO	FIT	SPAGHETTI

Table 6.2: Example sentences used in both training (top) and test (bottom) sets, across simple, right-branching, and centre-embedded sentence structures.

	Performance (GPE) (SD)		
	Simple	Right	Centre
32x32	0.017934 (0.013888)	0.012134 (0.00896)	0.009556 (0.00835)
20x20	0.00834 (0.001726)	0.015281 (0.025812)	0.028311 (0.031452)

Table 6.3: Mean grammatical prediction error (GPE) for Simulation 6.2 across simple, right-branching, and centre-embedded sentence structures for simulation layer sizes of 20x20 (n=12) and 32x32 (n=6). Values in parentheses represent the standard deviation.

Figure Captions:

Figure 6.1: A subset of activation maps after training for Simulation 6.1. Where layer 1 acquires unique sensorimotor representations for each concept, layer 2 acquires representations of grammatical category (noun, or verb).

Figure 6.2: Similarity maps of Simluation 6.1 for both layers 1 and 2. Regions representing individual nouns and verbs (layer 1), and the grammatical class of noun and verb (layer 2) are labbeled.

Figure 6.3: The results of Simulation 6.2 for a familiar centre-embedded sentence, "FROG - WHO - GIRL - PAINT - HUG - SPRINKLER". The input layer (layer 1) represents sensorimotor conceptual information, where layers 2-3 represent grammatical category. The output layer (layer 5) generates unique grammatical predictions for each transition.

Figure 6.4: The results of Simulation 6.2 for a novel centre-embedded sentence, "NAPKIM - WHO - SUN - BRING - WISH - APPLE". The activation patterns in layers 2-5 are nearly identical to those in Figure 3 (familiar sentence), indicative of a network that is strongly systematic.

Figure 6.5: A tagged association map for Simulation 6.2 Layer 5. Topographic clustering shows a global preference for sentence structure (simple, right-branching, or centre-embedded), with local clustering based on sequence element. Values in parentheses represent the index of a given part-of-speech, where colour coding reflects sentence structure.

Figure 6.6: Mean transitional prediction values across each possible transition in the van der Velde et al. grammar averaged across 12 simulations of network size 20x20. The network displays excellent performance across all transitions. Ungrammatical transitions are marked with an asterisk.

Figure 6.7: Mean transitional prediction values across each possible transition in the van der Velde et al. grammar averaged across 6 simulations of network size 32x32. The network displays excellent performance across all transitions. Ungrammatical transitions are marked with an asterisk.

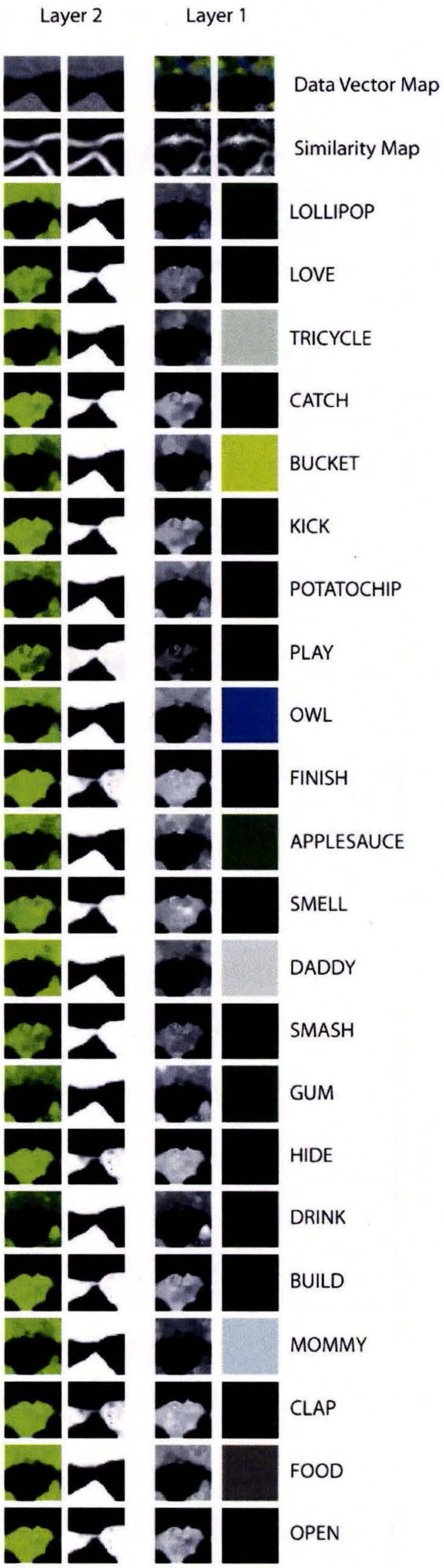

Layer 2
Layer 1
Data Vector Map
Similarity Map
LOLLIPOP
LOVE
TRICYCLE
CATCH
BUCKET
KICK
POTATOCHIP
PLAY
OWL
FINISH
APPLESAUCE
SMELL
DADDY
SMASH
GUM
HIDE
DRINK
BUILD
MOMMY
CLAP
FOOD
OPEN

Figure 6.1

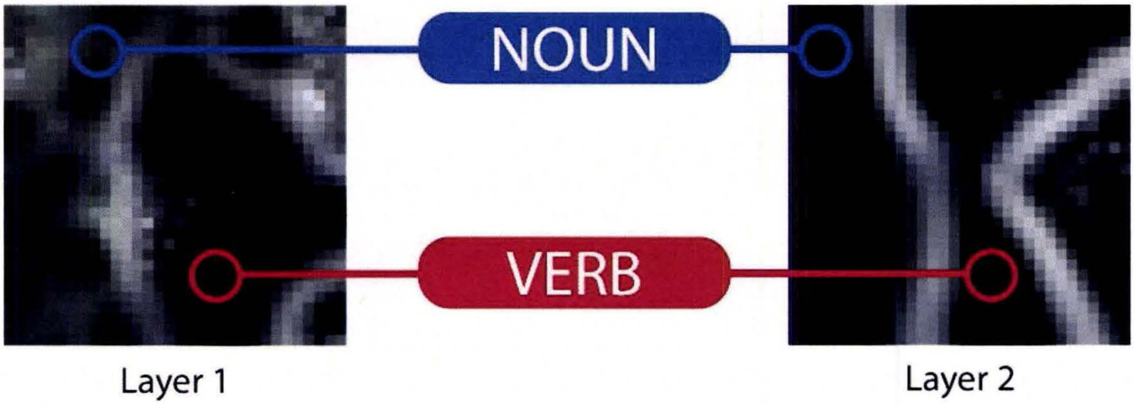


Figure 6.2

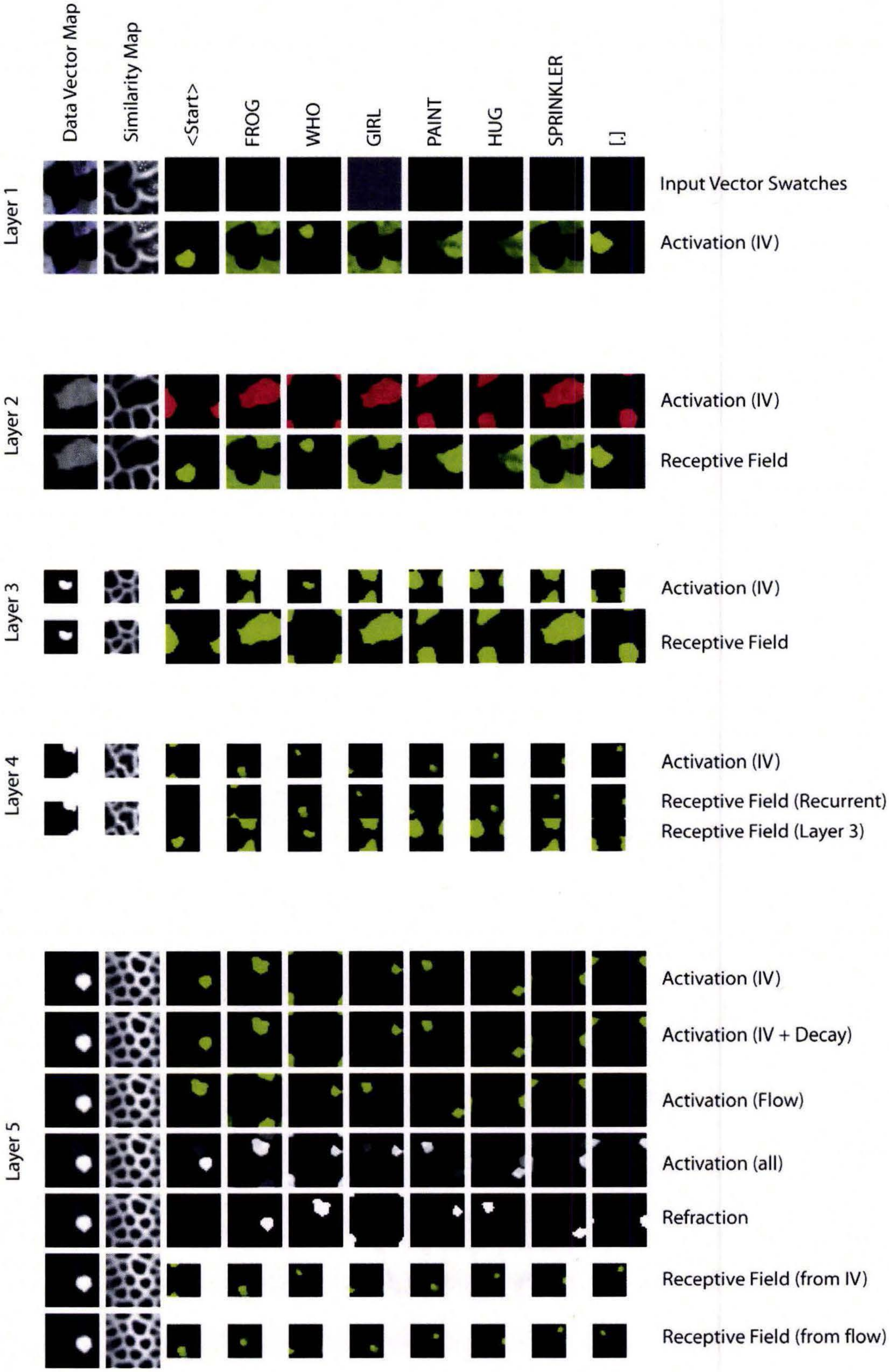

Data Vector Map
Similarity Map
<Start>
FROG
WHO
GIRL
PAINT
HUG
SPRINKLER
[.]
Layer 1
Input Vector Swatches
Activation (IV)
Layer 2
Activation (IV)
Receptive Field
Layer 3
Activation (IV)
Receptive Field
Layer 4
Activation (IV)
Receptive Field (Recurrent)
Receptive Field (Layer 3)
Layer 5
Activation (IV)
Activation (IV + Decay)
Activation (Flow)
Activation (all)
Refraction
Receptive Field (from IV)
Receptive Field (from flow)

Figure 6.3

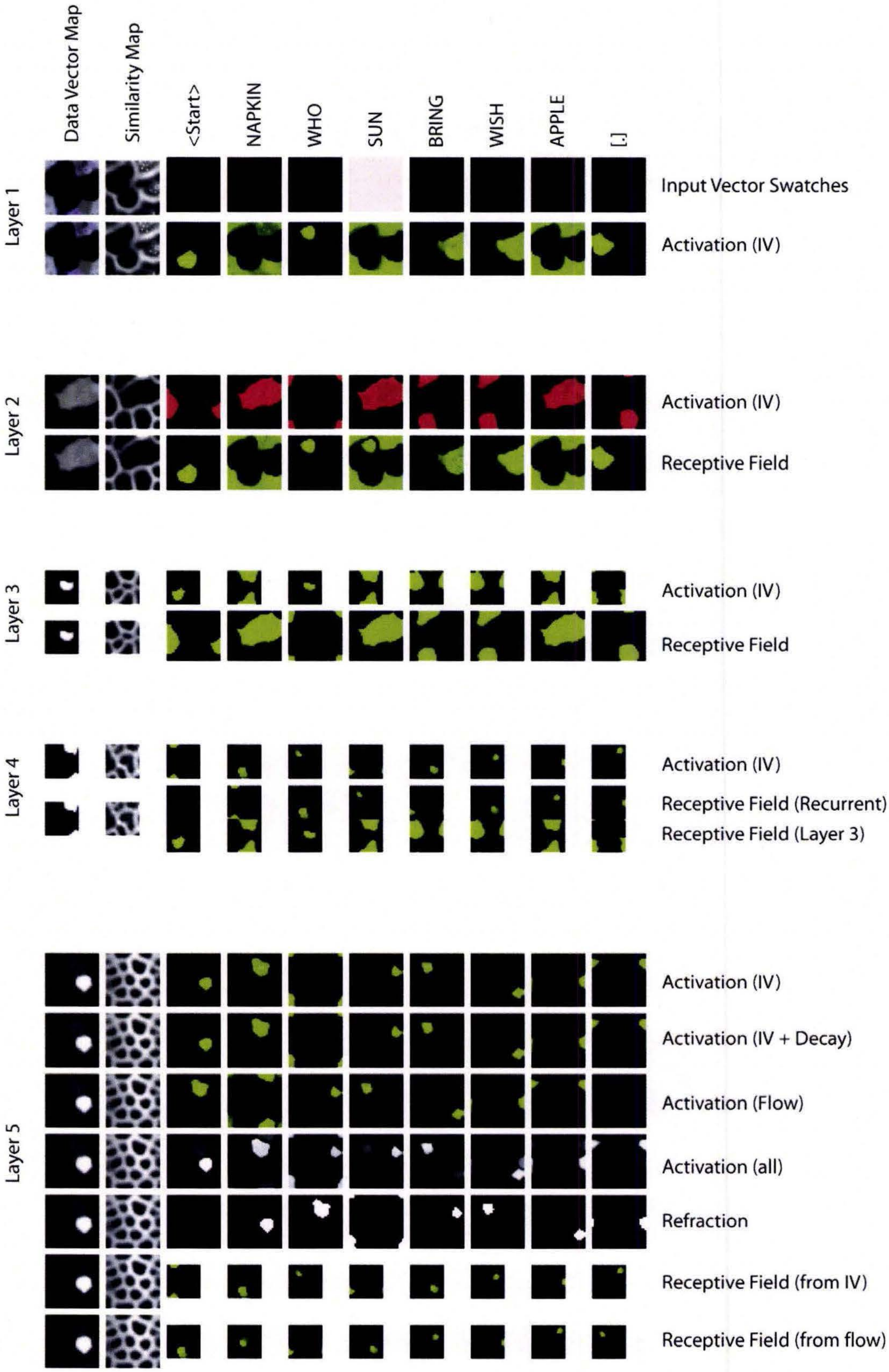

Data Vector Map
Similarity Map
<Start>
NAPKIN
WHO
SUN
BRING
WISH
APPLE
[.]
Layer 1
Input Vector Swatches
Activation (IV)
Layer 2
Activation (IV)
Receptive Field
Layer 3
Activation (IV)
Receptive Field
Layer 4
Activation (IV)
Receptive Field (Recurrent)
Receptive Field (Layer 3)
Layer 5
Activation (IV)
Activation (IV + Decay)
Activation (Flow)
Activation (all)
Refraction
Receptive Field (from IV)
Receptive Field (from flow)

Figure 6.4

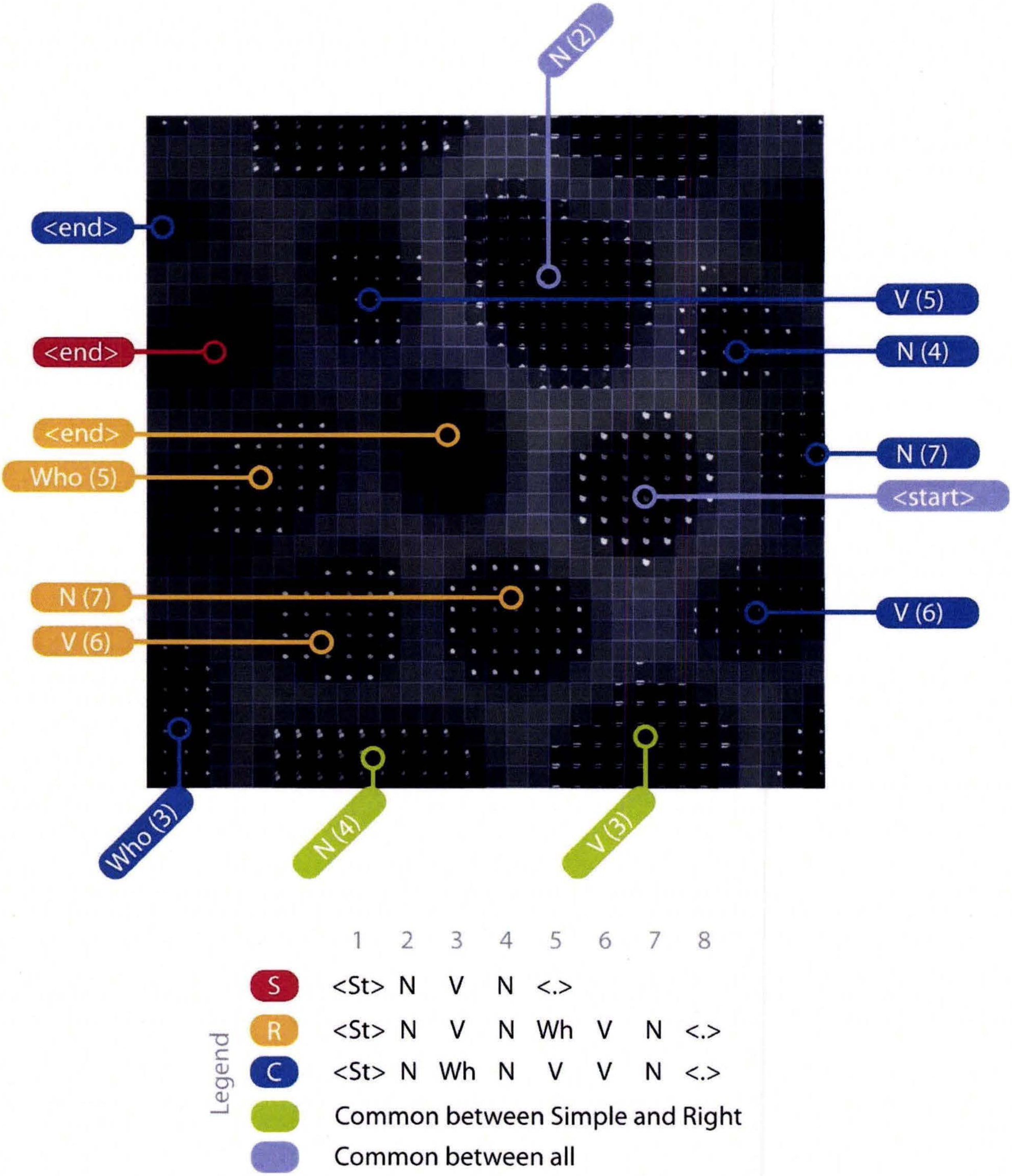

N (2)
<end>
V (5)
<end>
N (4)
<end>
Who (5)
N (7)
<start>
N (7)
V (6)
V (6)
Who (3)
N (4)
V (3)
1 2 3 4 5 6 7 8
S <St> N V N <.>
R <St> N V N Wh V N <.>
C <St> N Wh N V V N <.>
Common between Simple and Right
Common between all
Legend

Figure 6.5

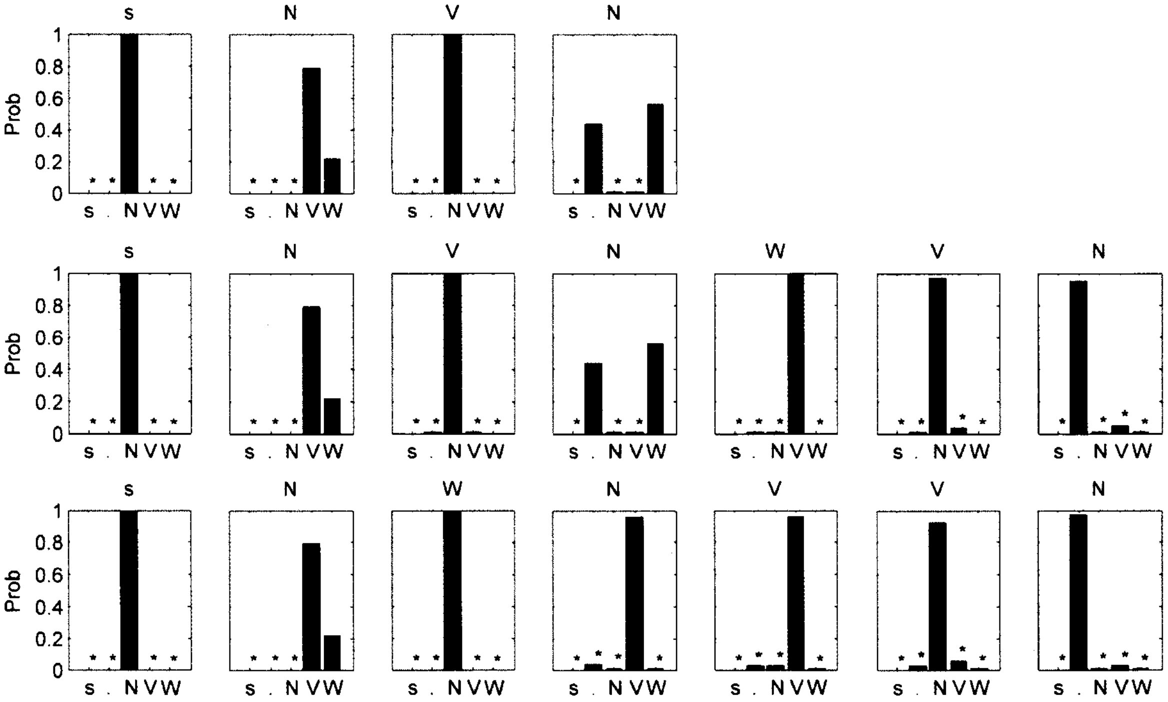


Figure 6.6

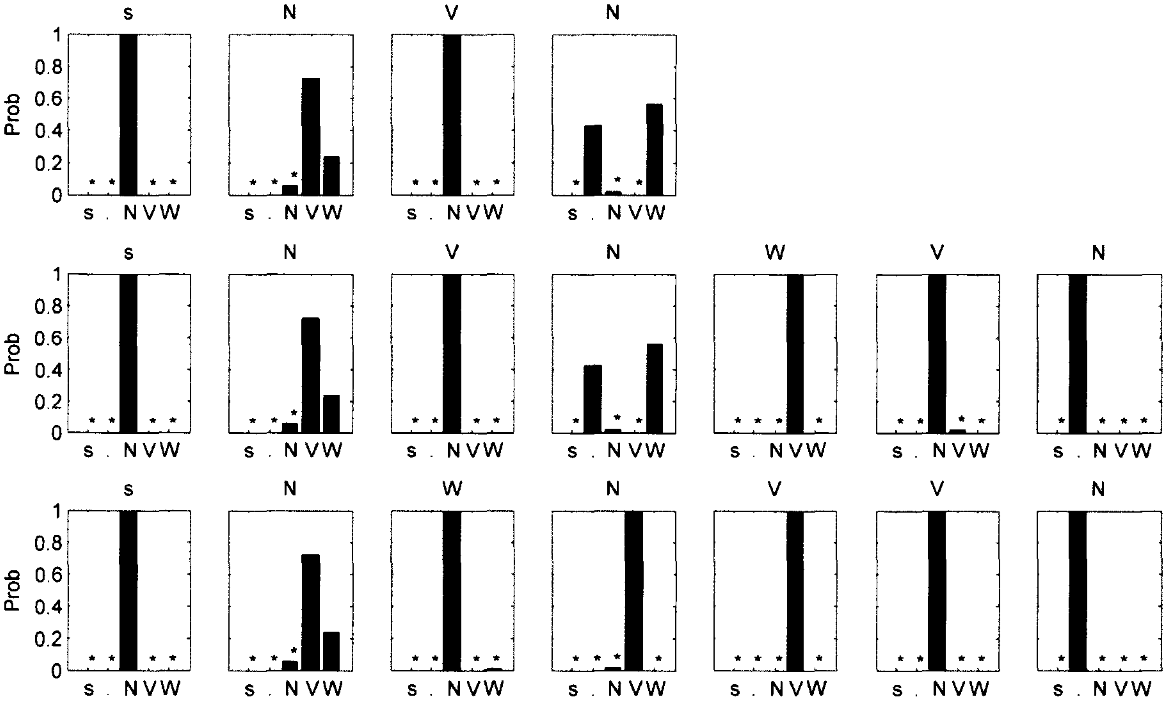


Figure 6.7

Chapter 7

Discussion

The Chimaera architecture was developed as a method of extending the self-organizing map to contain separate representations for conceptual sequence elements, as well as temporal information specifying grammatical sequences of those elements. This was accomplished by separating temporal information from both the SOM learning rule and the data vector representations of each node, in contrast to contemporary approaches to unsupervised temporal representation, and instead using a method based on intralayer Hebbian associative learning over time. This temporal flow produced predictive activation for the possible transitions from a given element to the next element in a sequence, but was unable to resolve transitional ambiguities. While a mechanism of interlayer feed-forward connection was able to add a restricted ambiguity resolution facility to the architecture, this mechanism required an implausibly large number of network layers to successfully acquire natural grammars, such as language. The Chimaera was migrated from a multilayer connection topology based on intra-layer recurrence, to a method based in inter-layer recurrence, similar in structure to both the simple recurrent network, and the RecSOM. While this method allowed the Chimaera to acquire even very deep transitional ambiguities in grammars, in terms of language modeling the Chimaera was still operating at the level of grammatical category rather than the level of the word. Incorporating an activation function based on a semantic co-occurrence metric allowed the Chimaera to accept grounded and developmentally plausible sensorimotor conceptual representations, and to abstract a general knowledge of grammatical category from these representations. With this transition from abstract symbolic to grounded sensorimotor input, the Chimaera was able to both operate at the lexical level, and to display excellent performance in a difficult test of grammatical systematicity on the benchmark van der Velde et al. (2004) grammar.

7.1 Comparison to Frank (2006) and Farkas & Crocker (2008)

The recurrent Chimaera model of Simulation 6.2 exhibits impressive performance on the next part-of-speech prediction task, generating ungrammatical predictions only 1.3% of the time, on average. This mean performance is approximately 3% better than the best performing unsupervised RecSOMsard model of Farkas and Crocker (2008), while near or above the performance of the best performing SRN and Echo-State Network models of Frank (2006) – though Frank uses an alternate measure of grammatical performance, so the measures are not entirely alignable.

The analysis of a network's performance beyond the grammatical prediction error is difficult. Frank (2006) argued that GPE, a measure of grammatical prediction performance, is a poor measure of grammatical systematicity in that it lacks a baseline measure, as a given transition may be more or less difficult than another, requiring differing depths of temporal look-back (and the systematic abstraction of lexical items required to obtain that look-back). In light of this, Frank (2006) introduced an alternate measure of performance, FGP, that evaluated grammatical prediction performance against a bigram statistical model that functions as a baseline. The issue with using a bigram model as a performance baseline is that the Van der Velde et al. (2004) benchmark grammar contains ambiguities easily visible to hand-inspection that are deeper than 1st order, and as such, are deeper than a bigram model can detect. This difficulty is likely why Farkas (2008) later transitioned to using a performance metric of systematicity based on network performance surpassing the best available n-gram Markov model that the experimenter could generate. However, this n-gram Markov model performance metric still does not seem an adequate measure of systematicity. Fodor and Pylyshyn (1988) described systematicity as a defining property of a cognitive system, and being more systematic than an alternative model does not necessarily mean that one is being generally systematic, or that systematicity is a general property of the system one is examining.

To determine the difficulty of each transition in the Van der Velde et al. (2004) grammar, a recursive depth parser was implemented, where for each transition in each of the three sentence structures in the Van der Velde et al. grammar, the algorithm determined the maximum depth of lookback required in order to successfully generate an unambiguous prediction for that transition. This provides a more accurate measure of the relative difficulty of each grammatical prediction. The results of this analysis are included in the appendix. As a side effect of this process is to determine *all* the transitions present in the grammar for each depth, and whether they are ambiguous transitions or not (i.e., they require looking back more elements to determine their next transition), these data are included as well.

The analysis shows that the Van der Velde et al. grammar contains several 2nd order ambiguities, specifically at the 4th transition in the right-embedded grammar ("<Who> – V"), and the 4th transition in the centre-embedded sentence ("<V> – V"). These transitions correspond to the worst mean FGP performance on both the ESN model at approximately 0.4 out of a possible 1.0 (Frank, 2006), as well as approximately 0.65 out of a possible 1.0 for the RecSOMsard model (Farkas & Crocker, 2008), while all three models tended to show decreased performance leading up to these transitions. Additionally, both the ESN and RecSOMsard models show a substantial dip in FGP performance to 0.6 and 0.75, respectively, for the first transition in both simple and right-branching sentences ("N—V") while showing top performance on this transition in centre-embedded centences ("N—Who"). While this transition is not ambiguous in that it does not depend on context (the transition is in fact*free*, and either way is valid), the networks handle this transition with a substantial decrease in performance.

This analysis suggests that for difficult grammatical transitions – transitions that require deep lookback – the previous models including the SRN, ESN, and RecSOMsard network are unable to successfully acquire one or both of the generalized part of speech information (distilled from statistical co-occurrence), or deep grammatical sequence information. The performance of both the ESN and RecSOMsard models on the free transition in both the simple and right-branching sentence structures further suggests that these models have difficulty acquiring a grammar that allows multiple transition paths, either in the aspect of this acquisition that involves distilling part of speech, or in the simultaneous pairing of this distillation process with storing grammatical sequence information.

In comparison to the supervised SRN and ESN models of Frank (2006), and the unsupervised RecSOMsard model of Farkas & Crocker (2008), the current recurrent Chimaera model generates nearly completely correct predictions across all transitions in the network, including these deeply ambiguious predictions. In this way the performance distribution of the recurrent Chimaera model reflects a different process than is present in the previous models – where the SRN, ESN, and RecSOMsard models generally show decreased performance in areas where difficulty increases, the recurrent Chimaera's performance distribution is functionally flat, and does not show any performance degradation across any transition in the network. This performance distribution reflects what one would expect from a network exhibiting general strong grammatical systematicity across virtually all input.

This conclusion is further reinforced by the difference in test sets used in the recurrent Chimaera model versus the previous models. Frank (2006) and Farkas & Crocker (2008) have taken care to assemble training data with specific instances of nouns and verbs that appear in only a single sequential location in the parse, where they then check to see if that word is also accepted in untrained locations during the test set – for example, the word "bottle" might only ever appear as a direct object in the training set, where it would then be tested in a subject position in the training sentences. The test set of the recurrent Chimaera model has taken this one step further – the words themselves, represented by unique sensorimotor concept representations, are novel. That is to say, the model is not testing the transfer of one word instance to other positions that share that word category – the model is testing the general acquisition of two separate types of knowledge – word category, and grammatical sequence learning – and seamlessly transferring the two to novel instances of words.

One might argue that the networks are performing different tasks, and as such the performance comparison is not entirely accurate. This is true. The models of Frank (2006) and Farkas and Crocker (2008) are attempting to ascribe grammatical category to completely ungrounded symbolic input, and are further attempting to learn the grammatical sequence structures present in that input. In this case it is not the *problem* that has been over constrained, but the *input* to the problem. Stochastic methods have seen remarkable success across a variety of domains in language processing (Cristianini, 2010), and indeed both the previous models as well as the Chimaera model are using stochastic methods based on co-occurrence to ultimately distill grammatical category information from a language stream – however the poverty of the stimulus in the symbolic case is extreme, as the case is such that no referent information is available for the symbols, such that each lexical representation contains zero useful information to generalize its part of speech – or any other property for that matter. Fodor and Pylyshyn (2008) did not intend for their description of systematicity to apply to an algorithm operating on a set of unrelated numbers – they were describing the behaviour of human cognitive systems – and when we supply our networks with a subset of the same conceptual knowledge that young infants who are beginning to learn language have access to (Mandler, 2004), the network acquires a general behaviour of systematicity. Following the logic of the semantic bootstrapping hypothesis (Pinker, 1984), we might posit that systematicity is not a property of the cognitive system alone, but rather (in terms of Fodor and Pylyshyn) it can be a property of the pairing of an appropriately sensitive computational/representational system with an informative input set, that further is specifically informative at each *level of representation* that one wishes to be systematically sensitive to.

7.2 Perceptual Symbol Systems and Grounding

One of the most common tools in Natural Language Processing is the analysis of large corpora of text, and these methods have enjoyed remarkable success across a variety of tasks from search engines (Langville, Meyer, and Fernandez, 2008) to machine translation (Lopez, 2008). While unannotated corpora contain bare text, and as such are not semantically informative at the level of the word without incorporating additional or prior knowledge, contemporary models of human knowledge representation suggest that the conceptual information stored in our brains is not amodal textual information, but rather has a much more perceptually grounded, modal character.

Perceptual Symbol Systems theory (Barsalou, 1999) is a modal description of cognitive knowledge representation based on a wealth of empirical evidence (e.g. Pecher, Zeelenberg, and Barsalou, 2004; Richardson, Spivey, Barsalou, and McRae, 2003; Solomon and Barsalou, 2004) suggesting that the knowledge stored in brains is stored not amodally, but rather specifically localized in the modality where that information was acquired. For example, our concept of dog might include the notion of what a dog looks like, and the sound a dog makes when it barks.

Perceptual symbol systems suggests that our knowledge of the visual features that are part of a dog are stored in the visual areas of our brain, while our knowledge of the sound of a dog bark is stored in our auditory cortex. These individual, modal representations are joined together in conceptual integration areas that allow (for example) the cuing of the concept of a "dog" to invoke these perceptual representations stored across the cortex.

A central aspect of the formulation of perceptual symbol systems theory is that the modal conceptual representations we store are not infrequently used, but rather they are the vehicle we use to think and complete a variety of tasks. The act of mental "simulation", which involves coupling these modality specific representations with their respective perceptual cortex, has been shown to actively generate modal mental imagery that is then used to solve even simple tasks, including property verification (Solomon and Barsalou, 2004). As such, detailed sensorimotor conceptual representations appear not just available to cognitive processing, but also the kind of representations we generally make use of as we process information. It then seems probable that at least some subset of the sensorimotor information contained within the Howell et al. (2005) input set is not only available to infants, but also the kind of information that they are actively acquiring and using to think about their worlds.

The idea of grounding the symbols used in amodal theories or input sets (Harnad, 1990) is not new, and the importance of grounded representations has been investigated from determining their utility in language acquisition (e.g. Howell et al., 2005), to debating the implications of ungrounded text processing systems on intentionality (Searle, 1980). It has been convincingly demonstrated that grounded representations are not required to obtain impressive performance on a number of specialized high-level tasks, where (for example) stochastic co-occurrence methods can achieve good performance and agreement with human experts on the task of essay grading (e.g. Landauer and Dumais, 1997). Here, we demonstrate that similar co-occurrence methods, in the form of the semantic feature co-occurrence metric, can achieve excellent performance on grammar learning when coupled with a grounded corpus.

In the terms of Barsalou (1999), the input layer of the recurrent Chimaera model is acting as a "conceptual integration area", where perceptual features across a variety of modalities converge into a single coherent representation. While the model does not contain separate modality-specific networks sensitive to particular aspects of perception – such as colour, shape, or auditory frequency – or have any mechanisms to implement anything like the simulation described in perceptual symbol systems, the current model is not incompatible with these ideas, and indeed the model begins to look something like the architectural sketches Barsalou describes with only a few additions. The model might be trivially modified to include modality-specific networks for the categories of perceptual features found in the Howell et al. (2005) input set, where the output of these networks would then converge upon the current integration area, creating – at least partially – an architecture similar to that described by perceptual symbol systems. However, this architecture would be as yet incomplete, as a critical aspect of Barsalou's (1999) theory is the idea of perceptual simulation and imagery, which the current model neither implements, nor speaks to. Because the model does not depend on input coming from separate perceptual areas in order to function – the semantic feature co-occurrence metric is able to group together features that co-occur within a single-layer network on its own – the "convergence zone" of layer one is broadly compatible with a variety of approaches to modal representation. Fully integrating this work into a specific simulable model of modal representation is left as an avenue of further research.

7.3 The Limits of Grounding and Semantic Bootstrapping

While we have demonstrated that grounded sensorimotor conceptual representations can greatly aid an unsupervised process of determining the grammatical category of nouns and verbs, how useful might this technique be for acquiring more complex natural language grammars? Language can generally be divided into classes of words that can be grounded in the world, including parts of speech such as nouns, verbs, adjectives (such as "blue"), and adverbs (such as "quietly"), as well as into classes of words that are not easily concretely grounded, such as clauses or conjunctions, that function to convey structural relationships in event representations. This class of "function" words can not be easily specified in terms of sensorimotor representations, and are instead thought to be acquired through the relationships they convey between agents (Bates and Goodman, 1997).

In terms of the current model, the model would be unable to acquire parts of speech that could not be specified in terms of distinct, non-overlapping feature representations. This is already the case for the "who" clause marker in the input set of Simulation 6.2, where this marker was artificially assigned an arbitrary value on a single non-overlapping feature dimension to ensure its categorization as a unique part of speech. Where semantic bootstrapping suggests that sensorimotor concept representations may enable the initial acquisition of grammatical knowledge, it is likely that this process is incomplete as a full specification of grammatical category learning, and must soon migrate to other methodologies – either purely stochastic, as in the case of Frank (2006) or Farkas and Crocker (2008), or a combination of stochastic and segregation methods working to ground higher-level knowledge of events with symbolic labels, perhaps through a process of self-supervised error-driven learning based on expectation failure (Schank, 1982).

While function words require a higher-level knowledge of event structure, the semantic co-occurrence metric begins to lose utility much sooner, potentially when children begin to acquire adjectives. Adjectives are perceptual modifiers, further specifying the features of a noun, as in the case of a "cuddly cat" or a "sour apple". Because adjectives work to specify a specific aspect of a noun – or, in terms of the sensorimotor concept set of Howell et al. (2005), provide a specific value along a given noun feature dimension – the current semantic co-occurrence metric would be unable to distinguish between nouns and adjectives. One might imagine that the current metric might be adjusted to include not only semantic co-occurrence, but also feature diversity, such that adjectives – containing a value on only a single feature dimension – might be teased apart from nouns or verbs, which tend to contain values across many sensorimotor feature dimensions. While the perceptual act of "notice" has been suggested as a method of acquiring conceptual information (Mandler, 2004), and taking stock of a difference does tend to focus our attention on a single aspect of a stimulus that disagrees with our expectations, it is not clear that this is the mechanism young infants learn to acquire a knowledge of adjectives, or that this method might be usefully extended to acquiring other parts of speech, and as such, is left for future work.

7.4 Applicability to other models and architectures

In spite of the ultimate limitations of the semantic co-occurrence metric in terms of acquiring functional parts-of-speech, the technique itself of using a variety of non-Euclidian energy functions is a general property of self-organizing map architectures (e.g. Heskes, 1999).

As such, this specific technique is not limited to the recurrent Chimaera model presented here, but rather is broadly applicable to virtually any recurrent SOM variant capable of transferring the information contained within an activation map generated using the semantic co-occurrence metric to superordinate processing layers. Further, this technique might be adapted to supervised architectures as well. The Echo State Network (Jaeger, 2003) and similar classes of architectures use a sparsely connected recurrent layer to undertake a process of "dimensionality reduction", in effect attempting to generalize the input set into groups. Brackel and Frank (2009) argue that something very similar occurs in the SRN during training, where the network progressively represents co-occurring words as similar states in the network's representational space. As Howell et al. (2005) found that rich sensorimotor concept representations improve grammar training performance in the SRN, it is possible that, over the course of training, the SRN is able to progressively acquire something very similar to a semantic co-occurrence metric, and to use this information to extrapolate at least some aspects of the underlying grammar in input sentences.

This brings to light a key difference between the current model and typical SRN models of grammar learning. These SRN models often attempt to acquire both a process of generalizing grammatical category information as well as a knowledge of deep grammatical structure concurrently in the same 3-layer network. At the heart of the matter is whether a given network (or subset of a network) is capable of acquiring more than one type of computational process. Recent supervised models of psycholinguistic production have had particular success taking a "dual-representational" approach, separating a knowledge of words and grammar into separate supervised networks (Chang, Dell, & Bock, 2006), and the recurrent Chimaera model presented here does something very similar. Layer one functions to acquire grounded sensorimotor conceptual representations, where layer two works to distill the commonalities in the representations of layer one into parts-of-speech, and layers three through five undertake a process of sequence learning that makes use of this grammatical category information to arrive at a knowledge of grammar. The issue with the performance of the supervised models of Frank (2006) may simply be that too many processes are being ascribed to a single network, where devoting a separate network to each level of representation present in the task of grammar acquisition from lexical sentence representations might see these supervised networks reach a level of performance similar to the recurrent Chimaera model presented here.

7.5 Unsupervised models

From the perspective of developmental plausibility, there is something appealing about an unsupervised model that achieves particularly good performance at the task of grammar learning. Fodor and Pylysyhn (1988) describe systematicity as a fundamental property of cognitive architectures, and in this respect the idea of training a network to be systematic by giving it prior knowledge of the regularities present in the world suffers from a variety of issues in terms of developmental plausibility, particularly in that it is unclear whether very young infants might be able sensitive to any error signals a teacher might provide without subscribing to an account of language learning that is heavily nativist (e.g. Chomsky, 1959). It has been suggested that if error-based learning is present at the earliest stages of language learning (but, for an argument against in the context of grammar learning, see Brown and Hanlon, 1970), that this error-driven learning might itself be unsupervised, or "self-supervised" (Elman, 1991) and driven by a process of expectation failure. In such a process, the young infant would generate predictions about the language stream, and in cases where these expectations failed to come about, the infant could learn something new about the use of language.

Even those widely invested in unsupervised learning would concede that at some point, infants begin to be sensitive to error signals, and are ultimately able to include a process of error-driven learning into their repertoire of techniques used to acquire knowledge about their worlds. But it follows that before a process of error-driven learning is available to the infant, they must make use of other, unsupervised techniques that do not depend upon both having *access* to, and being *sensitive* to an error signal. Hebbian learning (Hebb, 1949), abstractly, is a general process of co-occurrence (or correlational) based learning that is known to occur at the lowest levels of biological neurology. Hebbian learning describes a process where, in a case where two interconnected neurons simultaneously activate within some temporally-short window, the efficacy of the connection (or the weight) between those two neurons will tend to strengthen. Over time and with repeated exposure, those neurons will then acquire the natural co-occurrence statistics of whatever input set they are presented.

Because co-occurrence learning depends upon the activation level of a given set of neurons, as well as whether they are interconnected, and not upon a mechanism whereby the weights of neurons are sensitive to error signals (or even capable of propagating these error signals, to train deep layered networks, e.g. Rumelhart et al., 1986), the technique is unsupervised. In this sense, the semantic co-occurrence activation metric introduced into the current model is also unsupervised, and as such the sensitivity of the infant to the systematicity in the language stream is brought about by a nativist sensitivity to co-occurrence. But as unsupervised techniques do not necessarily imply biological or developmental plausibility, in what sense might this metric be considered cognitively plausible?

7.6 Plausibility of the semantic co-occurrence metric

Semantic priming (Meyer, Schvaneveldt & Ruddy, 1975) is generally considered a classic finding in cognitive psychology, and is the behavioural observation that humans tend to be faster at processing information across a variety of tasks when the information they are presented at a given trial is semantically related to information they have previously seen on a recent trial. Semantic priming can be contrasted to associative priming (Meyer & Schvaneveldt, 1971) or stimulus-response binding, in that where associative priming is purely correlational and can produce priming effects over repeated exposures of unrelated stimuli, semantic priming occurs specifically between semantically related stimuli, and occurs untrained, suggesting that it is a low-level property of the representational structure of the brain (e.g. Collins and Loftus, 1975; Collins and Quillian, 1969). As such, the semantic co-occurrence activation function used in the current model could be argued to use the very same semantic priming mechanism that we know occurs naturally in cognitive systems, albeit at a much higher rate of gain than conventional semantic priming. The semantic co-occurrence metric produces non-zero activation for any representation that shares features with another, and in this sense the intensity of the function is much higher than is traditionally ascribed in network models of semantic priming. While the difference in degree between the two may be a function of gain, it is also possible that the representationally coarse conceptual categories of the young infant (e.g. Eimas and Quinn, 1994) may make the handful of nouns and verbs they have acquired by age two (e.g. Nelson, 1973) seem far more similar than were their mental lexicon to contain the estimated 60,000 words of an average young adult (Bloom, 2000). As such, as a bootstrapping mechanism to begin to acquire semantic categories, semantic priming and the semantic co-occurrence metric may in fact be functionally the same process with the differences in degree an artifact of considering semantic priming on the scale and diversity of an adult conceptual system.

7.7 Plausibility as a temporal model of acquisition

While the current model achieves an impressive performance on the grammar learning task, the model makes several artificial assumptions about the temporal time-course of learning, and as such, is a poor model of the how infants acquire grammatical knowledge over time. The training of the model assumes that infants acquire a complete knowledge of their sensorimotor-grounded conceptual representational space before first beginning to distill a knowledge of grammatical category. Further, the model then assumes this knowledge of grammatical category must be complete and stable before beginning to acquire a knowledge of the valid grammatical sequences of parts-of-speech present in the input set, indicative of the grammatical rules of the sample language. Were any one of these assumptions not met, or the representations at any level modified after their initial training, the network would be unable to function. This temporal profile is of course dramatically different from that of infants acquiring language, where their knowledge of vocabulary begins initially quite slowly, before reaching a sudden spurt where the young infant begins to frequently and rapidly acquire new words (Bates and Carnevale, 1993). Concurrently, the infants knowledge of grammar and morphosyntax steadily increases in a more-or-less predictable fashion (Brown, 1973), where a knowledge of regular plurals tends to be acquired by 30 months of age, and before a knowledge of possessives at 34 months, which itself tends to be followed by a knowledge of regular past-tense at around 40 months old. This limit in the temporal profile of learning is not limited to the current model, but rather is a general limitation of virtually all contemporary neural network learning rules. This is generally referred to as the problem of catastrophic interference (e.g. Lewandowsky and Li, 1995), where periods of learning after initial training tend to erase or severely distort previous representations or processes learned by the network, resulting in a dramatic decrease or complete loss of performance.

One proposed solution to this problem is biologically inspired, from the process of neurogenesis. Neurogenesis refers to the observation that new neurons appear to be continually birthed in at least some areas of the cortex, and this appears critical to some aspects of cognitive function, including long-term memory (Becker, MacQueen, and Wojtowicz, 2009). Catastrophic interference might be mediated, at least in part, by novel learning algorithms that rely on adaptive mesh methods to selectively add new, trainable neurons into an expandable neural mesh that increases in size as a function of representational density. Architectures have already been proposed that make use of similar methods (e.g. see Kohonen, 1995b), and the successful adoption of similar architectures could allow future models to address what is currently a significant limitation in the utility of the model either in applied work, or as an investigative tool to model the interaction of the progressive acquisition of new conceptual and grammatical knowledge on the time-course of grammatical development.

7.8 Visualization and Transparency

Both the Chimaera architecture, as well as self-organizing models in general, are particularly amenable to the analysis and inspection of their internal representations and states without the use of complicated mathematical techniques. This has been a general criticism of connectionist approaches to modeling cognition (e.g. Fodor and Pylyshyn, 1988), where a lack of a deep understanding of the representational states and computational processes of a network functionally reduces the system to a black box, and as such the use of connectionist modeling in cognition can become something of a behavioural analysis of whole networks, rather than a reductionist understanding of their constituent parts. While a SOM does not use localist representations in the sense that they are used in supervised networks, nor could it similarly be said to use distributed representations in the same sense, one might consider a given vector – either input, or within a node – to contain a distributed representation of a given concept over a set of localist features. Similarly, while the patterns of activation produced by the network are neither purely localist nor distributed in the sense of Rumelhart et al. (1986), clusters of activation visible in the map do correspond to distinct representations or processing states, and the data vectors stored within the nodes of a given cluster can be easily analysed to produce a receptive field indicative of that node's contents in some (in this case) sensorimotor-conceptual or temporal processing-context space.

Further, the association map visualization provides an explicit visual representation of the associations between separate representational or processing states in a network layer, which can be used to infer temporal or grammatical chains along a set of input sequences. The sum of these visualizations provides a connectionist architecture that is remarkably transparent in every aspect, where these visualizations can be widely applied to tasks from broad analysis to illustrating network function for the purposes of teaching.

7.9 *Chimaera and summary*

The Chimaera network was developed as a vehicle to explore the simultaneous acquisition of conceptual and temporal representations, as well acquire processes that make use of those representations, in an unsupervised connectionist system. The architecture is based on the self-organizing map (Kohonen, 1982), itself a high-level abstraction of the topographic organization found in perceptual cortex. As such, one of the goals of this project was to take inspiration from temporal and multi-layered self-organizing map extensions and further specify the activation and associative dynamics of the network, such that the Chimaera might move closer to the ideal of a network specified completely at the level of a cellular automation, rather than a much higher abstract level. This goal was pursued in order to foster a deeper understanding of the extremely complex activation and flow dynamics involved in constructing a complete dynamic system specification of the low-level cellular-automata mechanics involved in arriving at a global system that exhibits the general behaviour of a temporal self-organizing map.

This development began with specifying the activation and flow dynamics for an intra-layer recurrent network that could produce useful temporal sequence predictions (Chapter 2), and lead to the integration and use of activation maps as temporal snapshots of a network's current "processing context", which could then be supplied to superordinate layers and used either as a feed-forward ambiguity resolution mechanism, or to find higher-order structures present in an input stream (Chapter 3). While this lead to a network architecture with intra-layer flow dynamics that approached the ideal of a cellular automation, these multi-layer feed-forward networks required an implausibly large number of layers to generate unambiguous sequence predictions for even moderately ambiguous sequences that analysis showed conventional supervised architectures using inter-layer recurrence were easily capable of processing (Chapter 4). Moreover, the Chimaera required substantially more computational power to simulate these flow dynamics, and as such was both slower to simulate as well as less capable at temporal representation than the most adept recurrent supervised architectures.

Inter-layer recurrent processing in the Chimaera was examined (Chapter 5), where a training scheme was first experimentally abstracted, and it was found the computationally intensive intra-layer recurrence processing was functionally redundant for all layers but the final (output) layer in an multilayer recurrent Chimaera. This functionally reduced these subordinate layers to a RecSOM with a specialized activation function, and eliminating this redundant processing allowed simulation efficiency to increase. The recurrent Chimaera was also found to be amenable to processing the benchmark language grammar of Van der Velde et al. (2004) as a test of systematicity, where the network was able to handle repeated sequence elements and deeply embedded ambiguities with ease, although extensions were required before the network could operate at the lexical level.

A method of distilling part-of-speech from semantic co-occurrence was developed (Chapter 6), where it was shown that operating on a corpus of grounded sensorimotor conceptual representations could greatly aid the unsupervised process of learning the grammatical category of nouns and verbs. Using an amalgamated Chimaera model, it was shown that a near complete knowledge of the Van der Velde et al. (2004) grammar can be acquired from sequences of grounded sensorimotor representations arranged in grammatical sentence structures, in support of the semantic bootstapping hypothesis of grammar acquisition (Pinker, 1984). This unsupervised model outperformed previous contemporary supervised and unsupervised models (Farkas and Crocker, 2008; Frank, 2006), and further displayed a functionally different pattern of results broadly across all transitions in the input set – namely, the general acquisition of the systematicity present in the grounded language stream.

Appendix: Recursive dependency analysis.

This analysis displays each of the transitions in the van der Velde et al. (2004) grammar, as well as the depth of each transition, and whether a given transition is ambiguous. (Note: The end marker is not included in this analysis.)

```
0:0: 'START'      0:1: TO 'N'       LENGTH = 2  **AMBIGUOUS**
1:0: 'N'          1:1: TO 'V'       LENGTH = 2  **AMBIGUOUS**
2:0: 'V'          2:1: TO 'N'       LENGTH = 2  **AMBIGUOUS**
3:0: 'N'          3:1: TO 'WHO'     LENGTH = 2  **AMBIGUOUS**
4:0: 'WHO'        4:1: TO 'N'       LENGTH = 2  **AMBIGUOUS**
5:0: 'V'          5:1: TO 'V'       LENGTH = 2  **AMBIGUOUS**
6:0: 'WHO'        6:1: TO 'V'       LENGTH = 2  **AMBIGUOUS**

7:0: 'START'      7:1: 'N'    7:2: TO 'V'       LENGTH = 3  **AMBIGUOUS**
8:0: 'WHO'        8:1: 'N'    8:2: TO 'V'       LENGTH = 3
9:0: 'N'          9:1: 'V'    9:2: TO 'N'       LENGTH = 3  **AMBIGUOUS**
10:0: 'V'         10:1: 'V'   10:2: TO 'N'      LENGTH = 3
11:0: 'START'     11:1: 'N'   11:2: TO 'WHO'    LENGTH = 3  **AMBIGUOUS**
12:0: 'V'         12:1: 'N'   12:2: TO 'WHO'    LENGTH = 3
13:0: 'N'         13:1: 'WHO' 13:2: TO 'N'      LENGTH = 3  **AMBIGUOUS**
14:0: 'N'         14:1: 'V'   14:2: TO 'V'      LENGTH = 3  **AMBIGUOUS**
15:0: 'N'         15:1: 'WHO' 15:2: TO 'V'      LENGTH = 3  **AMBIGUOUS**

16:0: 'START'     16:1: 'N'   16:2: 'V'   16:3: TO 'N'   LENGTH = 4
17:0: 'START'     17:1: 'N'   17:2: 'WHO' 17:3: TO 'N'   LENGTH = 4
18:0: 'WHO'       18:1: 'N'   18:2: 'V'   18:3: TO 'V'   LENGTH = 4
19:0: 'V'         19:1: 'N'   19:2: 'WHO' 19:3: TO 'V'   LENGTH = 4
```

References

Bailey, M., Briner, J., and Chamberlain, R. (1994). Parallel logic simulation of VLSI systems. *ACM Computing Surveys*, 26, 255-294.

Barsalou, L. (1999). Perceptual symbol systems. *Behavioural and Brain Sciences,* 22, 577-660.

Bates, E., and Carnevale, G. (1993). New directions in research on language development. *Developmental Review*, 13, 436-470.

Becker, S., MacQueen, G. and Wojtowicz, J. (2009). Computational modeling and empirical studies of hippocampal neurogenesis-dependent memory: Effects of interference, stress and depression. *Brain Research*, 1299, 45-54.

Benke, K., and Wittenburg, P. (1996). A Self-Organizing Neural Network Approach for the Acquisition of Phonetic Categories. *Proceedings of the 1996 International Conference on Artificial Neural Networks*, 881-886. Springer-Verlag.

Bloom, P. (2000). *How Children Learn the Meanings of Words*. Cambridge, MA: MIT Press

Bock, K., and Griffin, Z. (2000). The persistence of structural priming: Transient activation or implicit learning? *Journal of Experimental Psychology: General*, 129, 177-192.

Broadbent, D., E. (1958). *Perception & communication.* New York: Pergamon.

Brachman, R. J. (1977). What's in a concept: structural foundations for semantic networks. *International Journal of Man-Machine Studies*, 9, 127-152.

Brachman, R. J. (1979). On the epistemological status of semantic networks. In N. Findler, (Ed.) *Associative Networks: representation and use of knowledge by computers, 3-50.*. New York: Academic Press.

Brackel, P., and Frank, S. (2009). Strong systematicity in sentence processing by simple recurrent networks. In: N. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, 1599-1604. Austin, TX: Cognitive Science Society.

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence*, 47, 139-159.

Brown, R. (973). *A first language: The early stages.* Cambridge, MA: Harvard University Press.

Brown, R. and Hanlon, C. (1970). Derivational complexity and the order of acquisition in child speech. In R Brown (Ed.), *Psycholinguistics*. New York: Free Press.

Cansino, S., Williamson, S. J., and Karron, D. (1994). Tonotopic organization of human auditory cortex. *Brain Research*, 663, 38-50.

Chang, F., Dell, S., and Bock, K. (2006). Becoming Syntactic. *Psychological Review*, 113, 234–272

Chappell, G., and Taylor, J. (1993). The temporal Kohonen map. *Neural Networks*, 6, 441-445.

Choe, Y., and Miikkulainen, R. (2003). Contour Integration and Segmentation with Self-Organized Lateral Connections. *Biological Cybernetics*, 90, 75-88.

Chomsky, N. (1957). *Syntactic Structures*. The Hague: Mouton

Chomsky, N. (1959). A review of Skinner's *Verbal Beahavior. Language*, 35, 26-58.

Clark, A. (2008). *Supersizing the mind: Embodiment, action, and cognitive extension.* Oxford: Oxford University Press.

Collins, A. M., and Quillian, M. R. (1969). Retrieval time from semantic memory, *Journal of verbal learning and verbal memory.* 8, 240-247.

Collins, A. M., and Loftus, E. F., (1975). A spreading activation theory of semantic memory. *Psychological Review*, 82, 407-428.

Cristianini, N. (2010). Are we there yet? *Neural Networks*, 23, 466-470.

Deliyanni, A., and Kowalski, R. (1979). Logic and semantic networks. *Communications of the ACM*, 22, 184-192.

Eimas, P. and Quin, P. (1994). Studies on the formation of perceptually based basic-level categories in young infants. *Child Development*, 65, 903-917.

Elman, J. (1990). Finding structure in time, *Cognitive Science*, 14, 179–211

Elman, J. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7, 195-224.

Elman, J. (1995). Language as a dynamical system. In R. Port and T. van Gelder (Eds.), *Mind as Motion: Dynamical Perspectives on Behaviour and Cognition.* Cambridge, MA: MIT Press.

Farkas, I., and Crocker, M. (2008). Syntactic systematicity in sentence processing with a recurrent self-organizing network. *Neurocomputing*, 71, 1172-1179.

Fodor, J., and Pylyshyn, Z. (1988) Connectionism and cognitive architecture: a critical analysis, *Cognition*, 28, 3–71.

Fodor, J., and McLaughlin, B. (1990). Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work. *Cognition*, 35, 183-204.

Frank, S. (2006). Learn more by training less: systematicity in sentence processing by recurrent networks. *Connection Science*, 18, 287–302.

Goldberg, A. (1999). The emergence of argument structure semantics. In B. MacWhinney (Ed.), *The emergence of language.* New Jersey: Lawrence Erlbaum Associates.

Gruber, T. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Technical Report KSL93-04, Stanford University, Knowledge Systems Laboratory.

Hadley, R. (1994). Systematicity in connectionist language learning. *Mind & Language*, 9, 247–272.

Hadley, R., and Hayward, M. (1997). Strong semantic systematicity from Hebbian connectionist learning, *Minds & Machines*, 7, 1–37.

Hagenbuchner, M., Sperduti, A., and Tsoi, A. C. (2003). A Self-organizing Map for Adaptive Processing of Structured Data. *IEEE Transactions on Neural Networks*, 14(3), 491-505.

Harnad, S. (1990) The Symbol Grounding Problem. *Physica D*, 42, 335-346.

Hebb, D. O. (1949). *The Organization of Behaviour: A Neuropsychological Theory.* NY: Wiley.

Herrmann, M., Hertz, J., and Prugel-Bennett, A.1 (1995). Analysis of synfire chains. *Network: Computation in neural systems*, 6, 403-414.

Heskes, T. (1999). Energy functions for self-organizing maps. In E. Oja and S. Kaski (Eds.), *Kohonen maps*, 303-315. Amsterdam: Elsevier

Hillis, W. D. (1989). *The connection machine.* Cambridge, MA: MIT Press

Howell, S., Jankowicz, D., and Becker, S. (2005). A Model of Grounded Language Acquisition: Sensorimotor Features Improve Lexical and Grammatical Learning, *Journal of Memory and Language*, 53, 258-276.

Ida, S., and Makino, J. (1992). N-Body simulation of gravitational interaction between planetesimals and a protoplanet: I. velocity distribution of planetesimals. *Icarus*, 96, 107-120.

Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. In:

Becker, S., Thrun, S., Obermayer, K. (Eds.), *Advances in neural information processing systems*, Vol. 15, 593-600. MIT Press, Cambridge, MA.

James, D., and Miikkulainen, R. (1995). SARDNET: A self-organizing feature map for sequences. In: *Advances in Neural Information Processing Systems, vol. 7.* Cambridge, MA: MIT Press

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.

Kohonen, T. (1988). The "neural" phonetic typewriter. *Computer*, 21, 11-22.

Kohonen, T. (1991). Workstation-based phonetic typewriter. *In Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, 279-288. Princeton, NJ.

Kohonen, T. (1995). *Self-Organizing Maps*. Springer, Berlin.

Kohonen, T. (1995b). The adaptive-subspace SOM (ASSOM) and its use for the implementation of invariant feature detection. In F. Fogelman-Soulié and P. Gallinari (Eds.) *Proceedings of the International Conference on Artificial Neural Networks 1995*, Vol. 1, 3-10.

Landauer, T., and Dumais, S. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104, 211-240

Langville, A., Meyer, C., and Fernandez, P. (2008). Google's pagerank and beyond: the science of search engine rankings. *The Mathematical Ingelligencer*, 30, 68-69.

Lewandowsky, S. and Li, S. (1995) Catastrophic interference in neural networks: causes, solutions and data. In Dempster, F., and Brainerd, C. (Eds.) *Interference and Inhibition in Cognition*, 329–361. San Diego, CA: Academic

Li, P., Farkas, I., and MacWhinney, B. (2004). Early lexical development in a self-organizing neural network. *Neural Networks*, 17, 1345-1362.

Lopez, A. (2008). Statistical Machine Translation. *ACM Computing Surveys*, 40, 1-49.

Mandel, D., Jusczyk, P., Pisoni, D. (1995). Infants' recognition of sound patterns of their own names. *Psychological Science*, 5, 314-317.

Mandler, J. (2004). *The foundations of mind: the origins of conceptual thought.* New York: Oxford University Press.

Marcas, G. (1998). Can connectionism save constructivism? *Cognition*, 66, 153-182.

Mayraz, G., and Hinton, G. (2002) Recognizing Handwritten Digits using Hierarchical Products of Experts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 189-197

McCarthy, J. (1960). Recursive functions of symbolic expressions and their computation by machine. *Communications of the ACM, 3, 184-195.*

McClelland, J. and Rumelhart, D. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of Basic Findings. *Psychological Review*, 88, 375-407.

McClelland, J., St. John. M., & Taraban, R. (1989). Sentence comprehension: A parallel distributed processing approach. *Language and Cognitive Processes*, 4, 287-335

McRae, K., de Sa, V. , and Seidenberg, M. (1997). On the nature and scope of featural representations of word meaning. *Journal of Experimental Psychology: General*, 126, 99-130.

Meyer, D., and Schvaneveldt, R. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 90, 227–234.

Meyer, D., Schvaneveldt, R., and Ruddy, M. (1975). Loci of contextual effects on visual word-recognition. In P. Rabbitt (Ed.), *Attention and performance V*. London: Academic Press.

Miller, G. (1990). WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3, 235-312.

Minsky, M. (1954). *Neural Nets and the Brain Model Problem*. PhD Dissertation, Pinceton.

Minsky, M., and Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press

Minsky, M. (1975) A framework for representing knowledge. In P. H. Winston (Ed.), *The Psychology of Computer Vision*, 211-277. New York: McGraw-Hill

Mozer, M. (1994). Neural net architectures for temporal sequence processing. In: A. Weigend & N. Gershenfeld (Eds.), *Predicting the future and understanding the past* (pp. 243-264). Redwood City, CA: Addison-Wesley Publishing.

Nelson, K. (1973). Structure and strategy in learning to talk. *Monographs of the Society for Research in Child Development*, 38.

Newell, A. & Simon, H. (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall.

Pecher, D., Zeelenberg, R., and Barsalou, L. (2004). Sensorimotor simulations underlie conceptual representations: Modality-specific effects of prior activation. *Psychonomic Bulletin & Review*, 11, 164-167.

Pinker, S. (1984). *Language learnability and language development*. Cambridge, MA: Harvard University Press.

Prasada, S., and Pinker, S. (1993). Generalization of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8, 1-56.

Quillian, M. R. (1969). The teachable Language Comprehender: A simulation Program and Theory of Language. *Communications of the ACM*, 12, 459-476.

Richardson, D., Spivey, M., Barsalou, L., and McRae, K. (2003). Spatial representations activated during real-time comprehension of verbs. *Cognitive Science*, 27, 767-780.

Rosenblatt, F. (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, 65, 386–408.

Rumelhart, D., McClelland, J. , & the PDP research group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*. Volume I. Cambridge, MA: MIT Press.

Rumelhart, D. & McClelland, J. (1986). On learning the past tenses of English verbs. In D. Rumelhart, J. McClelland (Eds). *Parallel distributed processing: Explorations in the microstructure of cognition*. Volume II. Cambridge, MA: MIT Press.

Schank, R. (1972). Conceptual Dependency: A Theory of Natural Language Understanding, *Cognitive Psychology*, 3, 552-631

Schank, R. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. New York: Cambridge University Press.

Schank, R. & Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Earlbaum Assoc

Scholtes, J. (1991). *Kohonen feature maps in natural language processing*. Technical Report CL-1991-01, Institute for Language, Logic, and Information, University of Amsterdam.

Searle, J. (1980). Minds, brains, and programs. *Behavioural and Brain Sciences*, 3, 417-457.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J. L. (1989). Encoding sequential structure in simple recurrent networks. In D.Touretzky, (Ed.), *Advances in neural information processing systems I.* New York: Morgan Kaufman, 643-652.

Sharp, H., Long, L., and Kahn, M. (1990). Computational fluid dynamics on the connection machine. *Mathematical and Computer Modelling*, 14, 714-719.

Solomon, K., and Barsalou, L.(2004). Perceptual simulation in property verification. *Memory & Cognition*, 32, 244-259.

Strickert, M., and Hammer, B. (2003). Neural gas for sequences. In T. Yamakawa (Ed.), *Proceedings of the Workshop on Self-Organizing Neural Networks (WSOM)*, 53-58, Kyushu, Japan.

Ullman, M. (2001). The declarative/procedural model of lexicon and grammar. *Journal of Psycholinguistic Research,* 30, 37-69.

van den Bos, E., Poletiek, F. (2008). Effects of grammar complexity on artificial grammar learning. *Memory & Cognition*, 36, 1122-1131.

Van der Velde, F., Van derVoort van der Kleij, G., and De Kamps, M. (2004). Lack of combinatorial productivity in language processing with simple recurrent networks, *Connection Science*, 16, 21–46

Voegtlin, T. (2002). Recursive self-organizing maps. *Neural Networks*, 15, 979-991.

Von Neumann, J. (1945, 1993). First Draft of a Report on the EDVAC. Reprinted in: *IEEE Annals of the History of Computing*, 15, 27-75.

Vossen, P., P. Díez-Orzas, Peters, W. (1997). The Multilingual Design of EuroWordNet. In: P. Vossen, N. Calzolari, G. Adriaens, A. Sanfilippo, Y. Wilks (Eds.) *Proceedings of the ACL/EACL-97 Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for Natural Language Processing Applications*, Madrid, July 1997.

Wolfram, S. (2002). *A new kind of science.* Champaign, IL: Wolfram Media.

Woods, B. (1975). What's in a Link: Foundations for Semantic Networks. In D. Bobrow and A. Collins (eds.), *Representation and Understanding: Studies in Cognitive Science*, New York: Academic Press