The University of Queensland

School of Information Technology and Electrical Engineering

On the Origins of Linguistic Structure: Computational models of the evolution of language

by

Bradley Tonkes, B.Sc. (Hons)

Submitted for the degree of,

Doctor of Philosophy

on

October, 2001.

Statement of Originality

I hereby declare that the work presented in the thesis is, to the best of my knowledge and belief, original and my own work, except as acknowledged in the text; and that the material has not been submitted, either in whole or in part, for a degree at this or any other university.

Bradley Tonkes, B.Sc. (Hons)

Abstract

This thesis explores a perspective for explaining the origins of linguistic structure that is based on considerations beyond the constraints of the language acquisition device. In contrast to the theory of Universal Grammar proposed by Chomsky, this perspective considers how the processes of language acquisition and use create a dynamical system that is capable of adapting linguistic structure to the inductive biases of learners. In this view it is possible to conceive of language adapting to aid its own survival: those languages that are more reliably and easily acquired will tend to persist for longer than their less easily learned counterparts. Thus, linguistic structures are seen as emergent, adaptive phenomena rather than pre-ordained features of language.

The particular issue that this thesis investigates is the extent to which language adaptation can facilitate acquisition by *general-purpose* learners. In the Generative Grammar tradition much is made of the necessity for domain-specific constraints on the language acquisition device. (Indeed, that there must *be* a distinct mental component dedicated to language tasks.) This outlook is in contrast to the connectionist viewpoint, which posits far more moderately constrained, domain-general mechanisms. This thesis examines how language adaptation can give general-purpose, connectionist learners the appearance of being language-savvy learners.

A simulation framework is proposed in which agents attempt to communicate simple concepts to one another using sequential utterances. In earlier simulations we aim to maximise the learnability of a language for the communication task. Later simulations show how the processes of language production and acquisition, when iterated, are capable of *producing* such languages. In total, three series of simulations are performed.

The first series of simulations addresses the question of how linguistic structure adapts when sender and receiver disagree on the form of language that is easiest to learn. Analysis reveals that, if necessary, the structural properties of language can

take on forms that compromise between the competing constraints on sender and receiver.

The second series of simulations considers the bottleneck of linguistic transmission: the requirement that learners generalise from a limited set of observed utterances to the entire language. Results show that generalisability can be boosted in a naive, domain-general learner by allowing language to adapt to the inductive biases present in the learner.

The third and final series of simulations investigates how the dynamical characteristics of linguistic change depend on the properties that drive the dynamics. That is, we explore the range of conditions under which the iterated learning dynamic is sufficient to establish a learnable language throughout the population. The results of these simulations show that the iterated learning dynamic is indeed able to act as a generator of languages that general-purpose learners are capable of acquiring.

The results from these studies suggest that through the dynamics of linguistic transmission, language can adapt to the capabilities and biases of its users. Furthermore, that language can exploit the inductive biases of general-purpose learning mechanisms to facilitate their own acquisition, contrary to Universal Grammar's hypothesised need for an innate, domain-specific acquisition mechanism.

List of Publications

The following is a list of publications that were produced during the period of candidature. Publications that were based on the work appearing in this thesis have been highlighted ($\star$).

Tonkes, B. (1997). Simulation issues in spiking neural networks. In Dale, M., Kowalczyk, A., Slaviero, R., and Szymanski, J., editors, *Proceedings of the Eighth Australian Conference on Neural Networks*, pages 80–84.

Tonkes, B., Blair, A. D., and Wiles, J. (1998). Inductive bias in context-free language learning. In Downs, T., Frean, M., and Gallagher, M., editors, *Proceedings of the Ninth Australian Conference on Neural Networks*, pages 52–56.

Bodén, M., Wiles, J., Tonkes, B., and Blair, A. D. (1999). Learning to predict a context-free language: Analysis of dynamics in recurrent hidden units. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, pages 359–364. IEE.

$\star$Tonkes, B., Blair, A. D., and Wiles, J. (1999). A paradox of neural encoders and decoders, or, why don't we talk backwards? In McKay, B., Yao, X., Newton, C. S., Kim, J. H., and Furuhashi, T., editors, *Simulated Evolution and Learning*, volume 1585 of *Lecture Notes in Artificial Intelligence*, pages 359–364. Springer.

Tonkes, B. and Wiles, J. (1999). Learning a context-free task with a recurrent neural network: An analysis of stability. In Heath, R. A., Hayes, B., Heathcote, A., and Hooker, C., editors, *Dynamical cognitive science: Proceedings of the Fourth Australasian Cognitive Science Conference*, NSW, Australia. University of Newcastle.

$\star$Tonkes, B., Blair, A. D., and Wiles, J. (2000). Evolving learnable languages. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems 12*, pages 66–72. MIT Press.

Wiles, J., Shulz, R., Bolland, S., Tonkes, B., and Hallinan, J. (2001). Selection procedures for module discovery: Exploring evolutionary algorithms for cognitive science. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 1124–1129.

Wiles, J., Shulz, R., Hallinan, J., Bolland, S., and Tonkes, B. (2001). Probing the persistent question marks. In Spector, K., Goodman, E., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M., and Burke, E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 710–717, San Francisco, CA. Morgan Kaufmann Publishers.

Wiles, J., Tonkes, B., and Watson, J. R. (2001). How learning can guide evolution in hierarchical modular tasks. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 1130–1135.

⋆Tonkes, B. and Wiles, J. (in press). Methodological issues in simulating the emergence of language. In Wray, A., editor, *The Transition to Language*. Oxford University Press.

Acknowledgements

I am indebted to the many people who guided, supported, assisted, amused, motivated, distracted and entertained me during my candidature. Without their assistance, completion of this thesis would not have been possible.

First of all I would like to thank my supervisor, Janet Wiles, without whose expert guidance, feedback and all-round great supervising skills this thesis would not have been attempted.

Sincere thanks must go to Alan Blair who convinced me to persevere with the simulations when things weren't working. His innumerable suggestions of, "How about you try ... ," were of fundamental importance to the design of a working simulation system.

I am indebted to my family, John, Barbara, Elliot, and Catherine for their support, and for knowing when not to ask too many questions. I am also grateful to many of my fellow students: Rob, Steve, Jims, Chris, Kai, and James for their assistance, tangible or otherwise.

Finally, I would like to thank the School of Information Technology and Electrical Engineering (and its past incarnations as the Department of Computer Science, School of Information Technology, Department of Computer Science and Electrical Engineering, and School of Computer Science and Electrical Engineering) for their provision of computing facilities, accommodation and financial support which allowed me to travel to conferences both nationally and internationally.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human languages exhibit a phenomenal amount of complex internal structure. In written, spoken and signed forms of language, utterances can be decomposed into smaller components such as words or syllables which can in turn be decomposed into letters or phonemes. The ability to combine elements is one of the most distinctive characteristics of language. It allows the construction of arbitrarily many expressions through the combination of simpler elements that are finite in number. The simple elements can not be arbitrarily combined to form compound elements; human languages impose complex constraints on how and when elements can be combined. These constraints can range from the simple phonetic to the complex syntactic. The combination of these constraints serves to define the structure of a particular language.

Humans use a wide variety of languages, each of which has its own particular structure. However, the variety of structures found in human languages does not range arbitrarily: there appear to be limitations on the types of structures that human languages can utilise, so called linguistic universals. What is the source of these constraints on linguistic structure?

In the generative grammar framework pioneered by Chomsky (1957, 1965, 1981, 1986), the human learner is innately endowed with strong constraints on the types of linguistic structure that can be acquired. Such constraints are claimed to be necessary to make the task of language acquisition tractable. In the generative grammar framework, these constraints on human language learners effectively determine the constraints on linguistic structure.

The connectionist approach to understanding human linguistic abilities is a marked departure from generative grammar. The connectionist framework posits

that human learners use far more moderately constrained, domain-general learning mechanisms to acquire language. With connectionism, it is much less clear why languages should be constrained in form. One proposal is that the languages that are observed simply represent 'good solutions' to the problem of communicating complex meanings over serial channels. Hence, languages have many similarities because they use the same solution to the problem (Elman et al., 1996).

A comparatively unexplored notion is that factors *beyond* the innate constraints on the human language acquisition faculty can provide constraints on linguistic structure. A recent interest in this possibility has emerged in the community of researchers investigating computational models of the evolution of language (particularly the evolution of syntax). This body of work considers the emergence and subsequent evolution of language-like communication systems (i.e., communication systems that have some internal structure and that are qualitatively more complex than systems based on a finite repertoire of discrete signals). Since humans and human languages are incredibly complex, and because much of human linguistic history is unknown ('language leaves no fossils'), the focus of much of this work has changed from the explication of *human* linguistic evolution to understanding the *general principles* behind the evolution of language-like systems. Unlike the general principles of generative grammar which are based on the universal *properties* of language, the general principles here refer to the underlying, universal *mechanisms* that govern linguistic evolution.

The evolution-of-language approach lends itself to the perspective that language can be viewed as an adaptive, dynamical system whose characteristics are determined by the processes of language use and language acquisition. Thus, the pertinent question is how the properties of language use and acquisition affect the outcome of the dynamical system.

One major result has been that weak functional constraints on language acquisition can, over time, lead to the impression that languages are tightly constrained (Kirby, 1999a). In Kirby's model, populations of learners have a slight preference for particular language forms over alternative forms. After repeated generations of use and acquisition, the less preferred forms 'die out' in the population, leaving only the preferred form. An observer of the languages of the final population might conclude that the less preferred forms were not viable since no user ever employed them. Thus, the dynamical aspects of linguistic evolution makes a weak constraint on language use appear much stronger than it actually is. This form of result has

also been demonstrated in a connectionist setting (Batali, 1998).

A second principle that has been proposed as an underlying determinant of the dynamics of language change has been termed the 'bottleneck' of linguistic transmission (Kirby, 2000). Language passes from one generation to the next through the observed linguistic experience of the learner (as well as the innate constraints on the learner). A learner's observation of language is necessarily finite, whereas language (particularly in the case of human languages) can be infinite (in terms of the number of valid expressions). Thus, the learner must derive the information necessary for understanding and producing a limitless range of expressions from a limited number of examples: language is squeezed through the learning bottleneck.

This requirement on the learners to generalise has an accompanying effect on the set of viable languages: the languages themselves must be generalisable. To be generalisable, a language must have a predictable structure. If expressions and meanings are related arbitrarily then there is no basis on which to generalise. Thus, the structural properties of a language may in part be due to the requirement for generalisation (a property of the way that language is used) rather than the language faculty.

To demonstrate the bottleneck principle in action, Kirby (1999b, 2000, 2001) considered a computational model of a population of learners, attempting to communicate about a set of (structured) meanings. The members of the population started with no language and used random invention to bootstrap the system. In each step, a randomly chosen member was removed from the population and replaced with a new individual which was then trained on the language of the population from a limited number of examples (i.e., it was required to generalise). Kirby showed that over time the language of the population changed from having no structure (random associations between meanings and utterances) to being completely compositional (each component of the meaning could be identified as a component of the utterance). Thus, the structural properties of the language emerge as a result of the dynamics of acquisition and use rather than directly from the constraints on the processing mechanisms. Kirby's claim is that the learning bottleneck may act as one of the major factors in determining the structure of language, requiring that languages have a generalisable structure.

A third principle proposed in the literature concerns the roles played by representation and linguistic function. The style of representation in the generative grammar tradition is typically symbolic in nature, allowing an arbitrary relationship between

syntax and semantics (i.e., the semantic representation need not tightly constrain syntax because complex symbolic transformations can be applied). In contrast, the connectionist approach to language places significant emphasis on representation, which can radically alter the computational demands of linguistic tasks. In a series of 'a-life' simulations, Cangelosi (2001) considered the relationship between concept formation and the *function* of linguistic communication (i.e., what individuals were attempting to communicate about). Cangelosi demonstrated how individuals formed (semantic) representations that were closely related to how they interacted with the world. The nature of these representations consequently influenced the structure of the language that emerged. Studies by Steels (1997a) in an embodied cognition setting have shown related results. Thus linguistic structure may be partly determined by how individuals form representations through interactions with the world (i.e., the properties of the environment and sensors matter).

Many different computational models have been applied to studying the evolution of language and there is no single accepted methodology for investigating issues concerning the evolution of linguistic structure. Each different model has particular characteristics making it appropriate for studying particular phenomena. Given the youth of the field, and the breadth of unexplored issues, it is unsurprising that each researcher has typically developed their own model which has then been used to highlight a phenomenon of interest. The type of model typically reflects the researcher's background assumptions regarding the nature of human linguistic competence.

Much of the methodology developed in this thesis is motivated by a connectionist perspective on language acquisition. Thus, one of the major questions is the extent to which the adaptation of language can produce learnable, structured language from a general-purpose learner (i.e., one that has not been constructed with the specific intent of acquiring the skills to perform a particular task). Neural networks, particularly multi-layer perceptrons combined with backpropagation-of-error learning, are the most well-known approach to general-purpose learning. However, they are not well suited to the temporal (or sequential) mode of processing that seems critical for language. For these types of tasks recurrent neural networks are more suitable, having an ability to process data with temporal dependencies while retaining a general-purpose approach to learning. Consequently, the work presented in this thesis is based on issues that are derived from the decision to use recurrent neural networks as general-purpose learners in the context of the evolution of

language.

From the proposed methodology (presented in Chapter 3), the following issues arise.

1. The dependence of linguistic structure on a set of conflicting constraints on language acquisition, particularly between sender and receiver. Are aspects of the complexity of linguistic structure a consequence of adaptation to the *intersection* of a set of weak constraints?

2. How do the properties of the learning bottleneck influence the dynamics of linguistic evolution and the resultant linguistic structures? The learning bottleneck determines the nature of the generalisation requirements. As the generalisation requirements change, so should the emergent linguistic structure.

3. How do the principles underlying the evolution of language affect the dynamics of language change *in practice*? How does the nature of linguistic change depend on the particular instantiation of the underlying factors?

1.1 Thesis overview

This thesis explores a perspective for explaining the origins of linguistic structure that is based on considerations beyond the constraints of the language acquisition device. In particular, we consider the notion that the processes of language acquisition and use create a dynamical system responsible for the adaptation of language to the user and the emergence of linguistic structure. Our primary issue is the range of factors that can influence this dynamic. That is, the aspects of language acquisition and use that may be responsible for the emergence of linguistic structure in a dynamical context.

In **Chapter 2** we contrast two of the current frameworks for understanding human linguistic abilities, namely generative grammar and connectionism. The adaptation of language is proposed as an alternative theory and recent work within this field is reviewed. Particular attention is paid to computational modelling, particularly with respect to the syntactic features of language. Two models are reviewed in some detail — Kirby's (2000) Iterated Learning Model and Batali's (1998) Negotiation Model.

Following an examination of the issues involved in devising computational models, a methodological framework for studying language adaptation using recurrent

neural networks is proposed in **Chapter 3**. This methodological framework serves as the basis for all of the simulations presented in this thesis. The first of these simulations is presented in the remainder of Chapter 3 and considers the question of how linguistic structure adapts when sender and receiver disagree on the form of language that is easiest to learn. Analysis reveals that, if necessary, the structural properties of language can take on forms that compromise between the competing constraints on sender and receiver. Results from preliminary studies for this chapter were presented at the Second Asia-Pacific Conference on Simulated Evolution and Learning (SEAL98) and appear in the proceedings (Tonkes et al., 1999).

Chapter 4 considers the bottleneck of linguistic transmission: the requirement that learners generalise from a limited set of observed utterances to the entire language. It is human infants' amazing abilities to generalise human languages that motivates much of the theory of generative grammar, particularly the need for domain-specific, innate constraints on the learning mechanism. Consequently, in Chapter 4 we consider the extent to which generalisability can be boosted in a naive, domain-general learner by allowing language to adapt to the inductive biases present in the learner. Results show that, with successful adaptation, languages can facilitate significant generalisation. Furthermore, we consider how changes in the properties of the bottleneck (i.e., changes to the generalisation aspects of acquisition) change (a) the ability of the learner to perform the generalisation task, and (b) the structural properties of the emergent languages. This work was presented at the 1999 conference on Neural Information Processing Systems (NIPS*99) and appears in the proceedings (Tonkes et al., 2000).

The final set of simulations, presented in **Chapter 5**, extends the work of previous chapters to consider a *population* of language users. In this population of learners we investigate how the dynamical characteristics of linguistic change depend on the properties that drive the dynamics. These simulations take Kirby's Iterated Learning Model (reviewed in Chapter 2) and consider the generality of his results. That is, we explore the range of conditions under which the iterated learning dynamic is sufficient to establish a learnable language throughout the population. The results of these simulations show that Kirby's results can be replicated in an alternative domain, but that there is sensitivity to the range of parameters. Kirby attributed his results to the learning bottleneck. Chapter 5 concludes by questioning the necessity for the type of explicit bottleneck considered by Kirby, suggesting that the factors in his model that established structured languages can be substituted

by alternative mechanisms that preserve the dynamical behaviour of the system. The simulations in this chapter were presented at the Third Evolution of Language Conference (Paris, 2000) and will appear in a volume of selected works arising from the conference (Tonkes and Wiles, in press).

Chapter 2

Models of Language

2.1 The language puzzle

Humans are unique amongst species in their linguistic abilities. While many species have developed complex communication systems, none rival the intricacies of human languages. At the core of language's power is its compositional structure: the ability to construct utterances for novel, complex meanings by the combination of simpler parts. This capability has been referred to as, infinite expression with finite means.[1] Despite decades of research, the nuances of human linguistic abilities have evaded complete explanation. Several key questions remain active topics in the research community. The fundamental issues that remain unresolved include: how humans process language, how humans acquire language, why language is unique to humans, and why human languages takes on their particular forms. Given that these foundational aspects are as yet unexplained, it is unsurprising that there exist many competing frameworks in which language research is conducted. The most influential of these has been the generative grammar tradition of Chomsky, but the connectionist approach has gained substantial interest within the psycholinguistic community. The following sections briefly characterise and contrast these two approaches. A third alternative is described which is relatively recent and considers the evolution of language. It is this third alternative with which this thesis is most concerned.

[1] Most famously by Wilhelm Von Humboldt.

2.1.1 The generative tradition

Current linguistic theory is largely influenced by the generative grammar tradition established by Chomsky (1957) which has been refashioned over the past three decades (Chomsky, 1965, 1981, 1986).[2] The goals of this research program are twofold. The first goal is to describe a formal grammar that accurately reflects a language user's intuitions about the language. That is, to provide a grammar that accounts for primary linguistic data and which generalises in a way that a user of the language would expect. In Chomsky's terms, such a grammar attains *descriptive adequacy* (Chomsky, 1965). The second goal of this program is to provide a theory of why one (descriptively accurate) grammar provides a better account than another, thus providing *explanatory adequacy*.

The generative grammar tradition has been highly successful in explaining linguistic phenomena. Indeed, the field has advanced to such a state that an adequate explanation of even one of the current linguistic theories would require more space than is available here. However, the intricacies of generative grammar theories are not the primary issue. What we seek to do here is to expound the broader issues that underlie much of the work in this area so as to contrast generative grammar with the connectionist approach to language, described later.

In Chomsky's terms, a generative grammar is "simply a system of rules that in some explicit and well-defined way assigns structural descriptions to sentences" (Chomsky, 1965, p8). Significantly, Chomsky proposes that a generative grammar is what a language user *knows* about language (though such knowledge is not available via introspection). In doing so, he draws a fundamental distinction between *competence* and *performance*. Whereas analysis of linguistic performance involves the observable aspects of language (that is, concrete utterances), theories of linguistic competence are concerned with the underlying mechanisms of language use (a generative grammar). The hypothesis is thus made that at the core of language ability is an autonomous, modular competence that can be described by a set of discrete, formal rules.

Finding a grammar capable of assigning an appropriate structural description to the syntactic utterances of a language (and only the syntactic utterances) would fulfill the goal of descriptive adequacy. To attain explanatory adequacy, a theory must explain how language users acquire such a grammar. Such an explanation is

[2]Newmeyer (1986) provides an excellent historical account of the changes made to Chomsky's theories and the reasons behind these changes.

referred to as a theory of Universal Grammar (UG), and must be capable of accounting for cross-linguistic variation. In essence, a theory of UG lays the ground-rules for a generative grammar; it constrains the types of constructs that are available to a grammar, thus constraining linguistic variation. To summarise, the goals of linguists working within Chomsky's generative grammar tradition are: (a) to deduce grammars that describe languages appropriately, and (b) to find a grammar formalism for which there are appropriate mechanisms that explain language acquisition and account for language variation. Before describing the prevailing opinion on these two issues, it is helpful to consider some of the motivating issues.

The first important observation is the so-called poverty of the stimulus (Chomsky, 1965). The linguistic data that a human infant is exposed to — human linguistic performance — is notoriously noisy. Not only would many utterances be considered ungrammatical if closely inspected, but the learner is provided with no information as to which utterances are grammatical and which are ungrammatical. From this seemingly intractable position, children almost universally manage to acquire a language virtually identical to that of their parents. This feat is even more remarkable given Gold's (1967) analysis of the situation in terms of formal grammar induction. Gold considered the case of a learner trying to identify a language, L, from some class of languages, C, given examples of strings in the language. Gold proved that if the learner is presented with positive examples only (that is, only grammatical strings)[3], then the task is not possible for any superfinite class of languages, C. Chomsky (1956, 1957) had proposed complex grammar formalisms to describe human linguistic competence. He argued that grammars associated with finite-state mechanisms were inadequate, and instead advocated transformational phrase-structure grammars. Since grammar classes of this complexity are not learnable in the limit from only positive examples, linguistic theories had to provide a plausible explanation of how human infants could acquire language.

The observed critical period of language acquisition, whereby learners must be exposed to a language early in life for them to successfully acquire it, is another phenomenon that linguistic theories seek to explain. A third factor that has influenced the nature of linguistic theories is evidence of rapid creolisation (Bickerton, 1983), a phenomenon whereby children in a community whose adults are only able to communicate using a pidgin, develop their own language within a single genera-

[3]The situation for human infants who are presented with unlabelled grammatical and ungrammatical strings is even more difficult.

tion. That is, the children learn *language* from non-linguistic data. However, by far the most significant fact that a linguistic theory must explain is the ubiquity and uniqueness of language in the human species: every human group employs language, but no other species uses a system of communication that resembles language.

To address these issues, Chomsky (1965) posits the existence of an innately specified, domain-specific, modular language acquisition device (LAD) as a theory of UG. The proposed structural details of the LAD have varied over time (Chomsky, 1981, 1986), but the underlying theme remains the same. Essentially, the LAD consists of high-level constraints on the nature of generative grammar, such as those posited by the well-known X-bar theory (Jackendoff, 1977). A consequence of this theory is that language acquisition reduces to the problem of finding the appropriate settings of the high-level constraints that match those of the learner's community. The formal learnability arguments such as those of Gold (1967) are thus countered: the critical period can be explained as a 'switching off' of a disused module, and rapid creolisation is a plausible outcome of such an innate component. Furthermore, the problem of language capacity being a uniquely human characteristic is answered by claiming a unique endowment of an appropriate LAD. Cross-linguistic variation is merely a matter of the range of parameter settings allowed by the LAD. Similarities between languages are perceived to be a consequence of the same set of constraints imposed by the LAD.

While the above summary outlines the major premises of the generative grammar tradition, there is nevertheless much scope for debate, for example, on the precise constraints inherent in UG. Since this thesis focuses on an alternative approach to understanding linguistic competence, these issues will not be pursued. However, one area of significant contention that is of immediate relevance to the work in this thesis, is the origin of the LAD. The type of innately specified LAD proposed by Chomsky necessitates genetic specification. Chomsky has been wary of attributing this genetic specification to a gradual process of Darwinian adaptation, leaving some commentators (for example, Deacon, 1997), to label Chomsky's approach (perhaps somewhat unfairly) as a 'hopeful monster' theory.[4] Other researchers, most notably Pinker (1994; Pinker and Bloom, 1990), have argued that gradual, Darwinian evolution can account for the emergence of an innate linguistic competence.

[4]That is, a theory relying on a large-scale mutation to endow some hominid ancestor with a functioning LAD where none existed previously.

2.1.2 The dynamical hypothesis

Although the underlying foundations of the generative grammar tradition, sketched above, have been widely accepted, they are by no means the only framework for considering language abilities. The rebirth of connectionism in the mid-1980s, which owes much to the pioneering work of Rumelhart and McClelland and the PDP Research Group (Rumelhart and McClelland, 1986b; McClelland and Rumelhart, 1986), led to a reconsideration of the differences between symbols and (distributed) representations with intrinsic content. Compared with the limitations of 'brittle' symbolic representations, connectionist approaches offered the advantages of tolerance to noise and variation with graceful degradation. Many connectionist models also had the ability to learn from examples. Given its early successes, the connectionist approach was inevitably applied to the modelling of linguistic phenomena, including text-to-speech mapping (Sejnowski and Rosenberg, 1990), English verb morphology (Rumelhart and McClelland, 1986a; Hare and Elman, 1995), and simple grammars (Pollack, 1987; Cleeremans et al., 1989; Elman, 1990). However, the field has not been without sharp criticism from many in the symbolic tradition.

While the connectionist paradigm showed much early promise in a wide range of domains, a damning critique of its potential applications to language was published (Fodor and Pylyshyn, 1988), arguing that connectionism was incapable of supporting the recursive operations necessary for language, particularly systematicity and compositionality, without resorting to 'mere implementation'. That is, since neural networks are capable of simulating arbitrary symbolic automata (Siegelmann, 1993), it follows that they can *in principle* demonstrate both compositionality and systematicity. Fodor and Pylyshyn argued that such a solution would be uninformative, and insisted that connectionist automata were incapable of demonstrating these qualities using connectionist-style distributed representations. In the following years, much work focussed on disproving these claims (van Gelder, 1990; Chalmers, 1990; Christiansen and Chater, 1994; Hadley and Hayward, 1995), though it would be fair to say that to this day, no consensus has been reached (Jagota et al., 1999, present an interesting range of opinions). Nevertheless, studies of connectionist natural language processing (CNLP) have continued.

The introduction of the simple recurrent network (SRN) with its associated prediction paradigm has been particularly influential (Elman, 1990, see Fig. 2.1). In a surprising result Elman trained an SRN to predict lexical items generated by an artificial, simplified fragment of English grammar. After training, it was shown

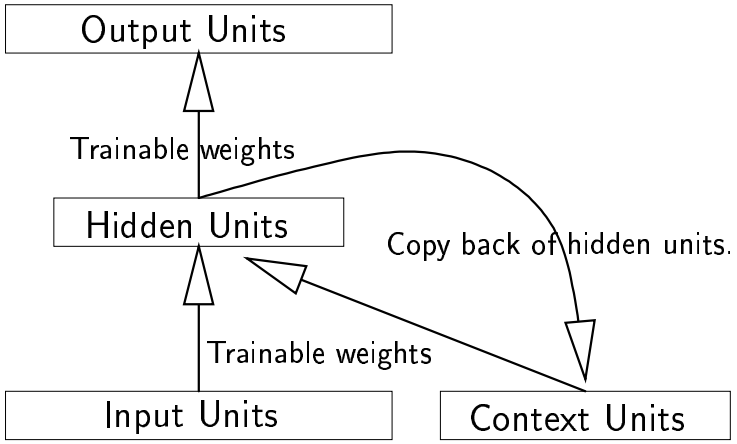


Figure 2.1: Elman's simple recurrent network (SRN; Elman, 1990). The SRN differs from feedforward neural networks in its ability to process temporal dependencies. At each time step the hidden units are copied back to the context layer. On the following time step, the context layer acts as an additional set of inputs for the hidden layer. Thus, the activations of the hidden units at time $t+1$ depend on the activations of the hidden units at time t. More precisely, for a network with I inputs, $in_0 \ldots in_{I-1}$, and H hidden units, $hid_0 \ldots hid_{H-1}$, the activation of the jth hidden unit at a time $t+1$ is given by $hid_{j,t+1} = f(\sum_{i=0,I-1} w_{i,j} in_{i,t+1} + \sum_{j'=0,H-1} w_{I+j',j} hid_{j',t})$, with some appropriate (nonlinear, differentiable) squashing function, f, and weight matrix, $\mathbf{w}$. Elman used this architecture with a prediction paradigm. A string of tokens, perhaps representing words or phonemes, is presented to the network one at a time. At each step the network is required to predict the next input token. That is, the input vector at time $t+1$ is the target output vector at time t.

that the SRN had derived syntactic categories such as noun and verb as well as finer grade semantic distinctions. Thus, syntactic categories were derived from the statistics of language usage, as found in the examples provided during training. The connectionist conception of linguistic ability represents a radical departure from the orthodox linguistic tradition outlined earlier. Instead of the Chomskyan notion of an autonomous language competence implementing grammatical rules and independent of the issues of language performance, there is the *dynamical* hypothesis (Pollack, 1991; Elman, 1995; van Gelder, 1998) which eschews the competence/performance distinction, and blurs the distinction between many of Chomsky's (1981) proposed language 'modules'.

> The idea is not merely that competence grammar needs to incorporate statistical and probabilistic information; rather, it is that the nature of language is determined by how it is acquired and used and therefore

> needs to be explained in terms of these functions and the brain mechanisms that support them. Such performance theories are not merely the competence theory plus some additional assumptions about acquisition and processing; the approaches begin with different goals and end up with different explanations for why languages have the properties they have. (Seidenberg, 1997, p1601).

Work on CNLP, by Elman and others, demonstrated that linguistic performance could result from a dynamical mechanism, and that significant aspects of language, such as syntax, could be acquired through statistical means (Elman, 1991; Weckerly and Elman, 1992; Christiansen and Chater, 1999; see also the recent special issue of *Cognitive Science*, edited by Christiansen et al., 1999 which shows the breadth of research in the area). The claim is *not* that human infants are *tabula rasa* learners, rather that the innate endowment required for linguistic competence is not necessarily as domain-specific, modular, or high-level as that proposed by Chomsky.

While the dynamical approach to language differs substantially in form from the symbolic approach, there is nevertheless a strong underlying connection between the two. Indeed, the complexity of tasks used in connectionist studies is typically described with respect to symbolic automata theory. It has been shown analytically that recurrent networks are Turing-equivalent (Siegelmann, 1993), reducing to deterministic finite automata (Casey, 1996; Maass and Orponen, 1998) or below (Maass and Sontag, 1999) in the presence of different types of noise (for related work, see Moore, 1998). More surprisingly, the results from empirical connectionist studies have provided evidence that recurrent networks are capable of *learning* to process a variety of regular languages (Cleeremans et al., 1989; Pollack, 1991) simple context-free languages (Sun et al., 1990; Wiles and Elman, 1995; Blair and Pollack, 1997; Rodriguez et al., 1999) and a simple context-sensitive language (Chalup and Blair, 1999), albeit with some difficulty on the more complex classes (see, for example, Bengio et al., 1994; Bodén et al., 1999).

These results from connectionist language processing appear *prima facie* to contradict the learnability results that influenced the development of generative grammar theory (e.g., Gold, 1967). The CNLP community has responded to this dilemma in a variety of ways. One counter-claim is that Gold's conception of learnability is inappropriate for the dynamical hypothesis; "... the child's task is learning to use language, not grammar identification" (Seidenberg, 1997, p1601), a view that is certainly in keeping with the rejection of Chomskyan linguistic competence as an

appropriate framework for understanding human language acquisition. In Gold's model of learnability, success is attained only when the precise target grammar has been identified; there is no margin for error. Thus, for the connectionist view of language acquisition, which focuses on language use rather than grammar identification and allows some deviation from the ideal target, Gold's learnability results do not strictly follow.

Notwithstanding arguments over the nature of learnability, it remains to be demonstrated how the CNLP approach can add appropriate learning constraints to satisfy Gold's conditions, just as Chomsky's UG constrains generative grammar. CNLP rejects the strong form of innateness proposed by generative grammar (Chomsky, 1981; Pinker and Bloom, 1990), citing the lack of evidence for such an innate LAD from either studies of neurological development or studies of genetic action (Elman et al., 1996; Deacon, 1997). Instead, CNLP researchers argue for much weaker constraints and re-emphasise the significant role played by learning.

> Innate capacities may take the form of biases or sensitivities toward particular types of information inherent in environmental events such as language, rather than *a priori* knowledge of grammar itself. (Seidenberg, 1997, p1603).

The 'starting small' hypothesis has been proposed along these lines (Elman, 1993).[5] The argument is that the immaturity of human infants, and the associated limitations on memory and processing, gives them a significant advantage in learning language by providing them with an appropriate bias. Unlike the generative grammar tradition, this 'innate' component is not domain-specific, but rather a general property of the cognitive system.

The dynamical approach to language has pushed for a reconceptualisation of the concept of innateness (Elman et al., 1996). Arguing from an ontogenetic perspective, Elman et al. suggest that claims of some (behavioural) traits being innate are imprecise, and that explanatory precision comes from describing *how* something is innate, that is, describing how the mechanisms of development lead to the observed characteristics. Only in this type of framework is it possible to understand the sympathetic roles played by nature and nurture.

[5] See also Rohde and Plaut (1997) for results that contradict those presented by Elman. Newport (1990) also offers a similar hypothesis to that of Elman.

2.1.3 The essence of the debate

Before continuing, it is worthwhile contrasting the relative positions of the two camps. Beyond issues regarding the *nature* of the language processing machinery (that is, whether it is best characterised as a symbolic or a sub-symbolic system), the most important difference between the generative grammar tradition and the connectionist approach is the question of innateness. The Chomskyan tradition argues for a domain-specific, modular language competence (UG) which provides much of the necessary grammar processing abilities and which reduces the problem of language acquisition to the setting of high-level parameters corresponding to the appropriate language. In contrast, the connectionist paradigm emphasises the significance of the role played by learning, claiming that statistically based learning, combined with much weaker domain-general biases, is sufficient for explaining language acquisition.

It is fair to say that at this time, models in the generative grammar tradition are capable of processing a wider range of syntactic constructions than connectionist models, a not unexpected result given the relative maturity of the two fields. Nevertheless, for those who doubt the plausibility of a 'language instinct', connectionism provides an attractive alternative. Two reasons are apparent for the connectionist approach's popularity amongst psycholinguists: a focus on the co-dependencies between development and learning; and a willingness to consider the knowledge that may be gleaned from the statistical properties of the linguistic environment. Indeed, it is the perception of the *cognitive plausibility* of CNLP that has won many proponents.

The question remains as to how appropriate learning constraints can be added to connectionist models, and the types of constraints that are necessary for successful learning. That is, an open issue is how (and what) prior knowledge should be incorporated into connectionist models. Certainly, connectionism is opposed to the types of explicit knowledge posited by generative grammar and instead considers knowledge in the form of biases arising from development (for example, Elman, 1993). This thesis considers an additional factor that might explain children's remarkable adeptness for language acquisition: that languages themselves adapt to be learned by human infants (Christiansen, 1995). Just as human infants may have some form of prior knowledge about the languages to which they will be exposed, languages may incorporate prior knowledge about the human language learning mechanisms.

Working within the connectionist paradigm, we aim to test the conditions under which aspects of languages can adapt to the weak biases of a learner, thus giving

the impression of a learner with innate linguistic expertise.

2.2 Modelling the emergence of language

The question of language emergence offers an alternative approach to understanding human linguistic competence. What motivates us to look at this field is the hypothesis that language adapts to aid its own survival. As other authors (Christiansen, 1995; Deacon, 1997) have pointed out, this hypothesis is not entirely new; Darwin (1890) for example, raises the possibility. Simply put, this hypothesis proposes that languages themselves can be seen as adaptive systems that are subject to selection pressures imposed by their human users. In Christiansen's view, "natural language is akin to an organism whose evolution has been constrained by the properties of human learning and processing mechanisms" (Christiansen, 1995, p9).

The consequence of this view is evident when placed in contrast with the typical portrayal of the problem. Rather than asking how humans manage to process language, the suggestion is that we instead ask how languages have adapted to be processed by humans. Of course, the absence of language in other species indicates that humans must have *some* innate biological capacity that enables language use. The question is the extent to which humans' capabilities are innate. Section 2.1 discussed the dichotomy between linguists in the generativist tradition of Chomsky who argue for an innate, domain-specific language module that reduces language acquisition to the setting of language parameter 'switches', and the connectionists who contend that human learning utilises the statistics of the information available in the environment. The adaptation of language provides an additional mechanism through which naive learners may give the appearance of expertise.

The study of language emergence has something of a blighted history. The speculative nature of much work in the field culminated in the infamous 1866 decision by the *Société Linguistique de Paris* to ban publication of papers on the topic. Nevertheless, the field is enjoying renewed interest as evidenced by the growing popularity of conferences in the area, such as the successful 'Evolution of Language' conference series (Hurford et al., 1998; Knight et al., 2000). The study of language origins is, necessarily, a highly cross-disciplinary field, drawing from anthropology (Noble and Davidson, 1996), neurology (Deacon, 1997), primate studies (Savage-Rumbaugh and Lewin, 1994), creolisation studies (Bickerton, 1983; Senghas, 1995), computational modelling (Hare and Elman, 1995) and evolutionary computation

(Kirby, 2000; Batali, 1998).

However, in this thesis we want to take a computational perspective on language adaptation. We are not so much concerned with knowing how human languages *in particular* came to be, but the necessary (computational) prerequisites for language-systems *in general* to emerge. That is, we are not so much focused on theories such as 'language evolved in response to increased socialisation of human ancestors' as we are in knowing what computational changes (which may or may not have been a consequence of socialisation) resulted in language emergence. The area of interest is the computational aspects of language emergence; genetic, behavioural, social and environmental aspects will be considered only in terms of their computational consequences. In the future it may be possible to more generally incorporate such broader factors into computational models, but at this stage modelling is not sufficiently sophisticated.

It is important, at this point, to draw a distinction between *language* systems and *signalling* systems. The defining distinction between language and signalling systems is that the utterances of language systems are structured and decomposable. That is, language utterances are compositional in nature. In contrast, the signals produced by a signalling system are atomic — there is no way to analyse an utterance by breaking it into smaller parts. The consequences of this distinction are profound. To acquire a signalling system an individual must be exposed to every possible utterance. If the population needs to communicate N meanings, they must employ N distinct signals. Alternatively, with language systems, individuals need only be exposed to some set of discrete tokens and acquire some means of assembling these tokens into complete utterances. The number of communicable concepts can then grow combinatorially with the number of atomic tokens and rules for assembly.

The primary issue addressed by computational simulations of signalling systems is the necessary and sufficient conditions for populations to converge on expressive systems. Expressive systems are those in which it is possible to unambiguously express each meaning; that is, one without homonyms. Such a system provides maximum communicative benefit. A common model for these simulations is one that provides a mapping between signals and meanings via a lookup table (for example, Lewis, 1969; Hurford, 1989). Early simulations established that it was possible for expressive systems to emerge as a result of either genotypic evolution or learning (for example, MacLennan and Burghardt, 1994; Werner and Dyer, 1991). However, due to the large number of interacting factors, the analysis determined neither necessary

nor sufficient conditions. A sufficient condition was established by Oliphant and Batali (1996) who proposed a learning procedure called the *obverter* and showed that if every agent in a population uses this procedure, then convergence on an expressive language is guaranteed. The obverter algorithm requires that agents choose to send the signal that has the maximum probability (averaged across the population) of being understood. This work has been extended to show how a variety of algorithms, many of which are biologically and socially plausible approximations to the obverter procedure, also guarantee convergence, and the relative rates at which each converges (Oliphant, 1999).[6] The other major focus of research in this area examines the necessary *social* conditions for signalling systems to successfully emerge, for example if it is necessary for the sender or receiver to attain some benefit from communication (Batali, 1995). Given the highly social context in which human language occurs, these issues are certainly significant from a language perspective. However, basic issues in creating a viable computational model of language need to be addressed before many of the broader social factors can be incorporated. It is these basic computational issues (such as the necessary types of learning algorithm, how individuals are rewarded for successful communication, and the length of time learners must be exposed to the language of the community) with which this thesis is concerned.

Three aspects of language have been examined using computational modelling of language origins: phonetics and phonology, the lexicon, and syntax (Steels, 1997b). Work in the first area, that of phonetics and phonology, considers the emergence of perceptual and articulatory systems that facilitate the recognition and production of distinctive vocal features. These issues are outside the scope of this thesis.

The second area of work, that of lexicon formation, explores the formation of the relationship between form and meaning (the symbol grounding problem) and how a population comes to agree on this relationship.[7] Several different computational frameworks have been proposed to study these issues, ranging from abstract simulation (Hutchins and Hazlehurst, 1995) through 'artificial life' worlds that share some characteristics with real environments (Cangelosi and Parisi, 1998; Cangelosi, 2001), to agents embodied in robotic implementations (Steels, 1996). Of this work, the role that symbol-grounding plays in determining utterance structure in Can-

[6] Interestingly, the obverter procedure seems to be particularly important for establishing expressive language systems, a point that we will return to later in §2.2.1.

[7] In this sense, studies of lexicon formation have significant overlap with studies of signalling systems discussed earlier.

gelosi's (2001) work, while not of direct relevance, makes an interesting point for this thesis. In his simulations, a population of neural network agents forage in a simulated ecology for mushrooms that vary in size and may be poisonous. Agents try to categorise the mushrooms they encounter and communicate information about the mushrooms to other agents. Significantly, the symbol-meaning associations on which the populations converge are related to the categorisation task that the agents must perform. That is, the representations that the agents form as a result of the categorisation task (as dictated by the external environment), directly influence the symbol-meaning association formed by the agent, and in later simulations, affect the way that the agents generalise utterances for novel environmental stimuli.

Some researchers modelling in the third area of language emergence, syntax, tend to regard lexicon formation as a secondary issue, and one which can be divorced from the problem of syntax (though notably, not Steels, 1997a). Cangelosi's results indicate that the lexicon may play a more significant role in the emergence of syntactic structures, for the reason that the formation and structure of the meaning domain will have a direct influence on the nature of the relationship between meanings and utterances. While these results are not of direct relevance, they provide a warning: that the choice of semantic domain and agent is not an arbitrary one, and that we should expect different languages to emerge for different domains.

The third area of work, the emergence of syntax, is most related to this thesis and is reviewed in greater detail.

2.2.1 Modelling the emergence of syntax

Two classes of frameworks can be distinguished amongst computational models of the emergence of syntax, identifiable as *macro-evolutionary* and *micro-evolutionary* models (Briscoe, 2000). Macro-evolutionary models treat language as an abstract entity. Rather than modelling language as discourse involving specific utterances, it is instead viewed as a set of parameters or features.[8] Macro-evolutionary models also often assume idealised conditions such as infinite populations and non-overlapping generations for analytic tractability (Briscoe, 2000). Using this type of model, Nowak et al. (2000) explored the evolutionary dynamics of the transition from non-syntactic to syntactic communication, showing that syntactic communication is advantageous only when the number of required unique signals exceeds some threshold. Other

[8]This view is compatible with the generativist linguistic tradition.

macro-evolutionary models have explored dynamics of language change (Niyogi and Berwick, 1995a,b), giving insights from dynamical systems theory into the reasons underlying historical changes in human languages. Most relevantly, Kirby (1998, 1999a) presents a macro-evolutionary model where linguistic universals emerge as a result of differential learnability of linguistic structures. This result is particularly interesting in terms of the debate on language innateness and specificity outlined in §2.1. It demonstrates that linguistic universals, often cited as evidence for a Chomskyan LAD, can emerge from the dynamics of *social* interaction and do not require genetic specification. This theme of dynamics of social interaction is one which runs through most simulations of the emergence of language, including those presented in later chapters.[9]

In contrast to macro-evolutionary approaches, the micro-evolutionary approach models language emergence on the level of individual utterances from specific members of a finite, heterogeneous population.[10] The focus of this thesis is how specific learning mechanisms influence language emergence, making a micro-evolutionary style model the appropriate choice of model class. The most popular approach has been to model the individuals of a population as symbolic grammar systems (Hashimoto and Ikegami, 1995, 1996; Briscoe, 1998; Kirby, 2000; Batali, in press), staying within the bounds of orthodox linguistic approaches to language production and comprehension. Batali's early work using neural network based agents is a notable exception (Batali, 1998). This distinction is an important one given the discussion of §2.1, in that the particular choice of approach significantly impacts on the types of simulations that can be performed.

There is much similarity in the basic premise of all of these micro-evolutionary simulations. A population of agents is created without a co-ordinated language. Each agent has its own mechanism for producing and understanding utterances; I-language in Chomsky's (1986) terms. During the course of the simulation, agents contribute utterances to an arena of use (E-language), which other members of the population try to understand and learn from. The changes resulting from learning affect the future utterances of the learner. Consequently, a complex dynamic is established between acquisition and production, shown in Fig. 2.2.

[9]Note that the issue in these studies is not the specific social context (e.g., that there were small groups of nomadic hunter-gatherers), but the more general outcomes of such an environment (e.g., that learners are exposed to language produced by a restricted subset of the population).

[10]That is, a population of simulated entities that varies across some set of parameters be it the weights of a neural network or the rules in a grammar.

The issues addressed by work on micro-evolutionary models of syntax emergence concern the conditions under which this dynamic can lead to stable, expressive language systems, including the types of phenomena observed in human languages. Hashimoto and Ikegami (1995) show how this dynamic can lead to an increase in grammar complexity (as measured with respect to the levels in the Chomskyan hierarchy) over time. However, with increased grammar complexity comes an increased parsing cost and populations may converge on a less-complex grammar. The languages evolve under a trade-off between expressivity and tractability — the benefit that a highly expressive language confers must be balanced with the cost of acquiring and using that language.

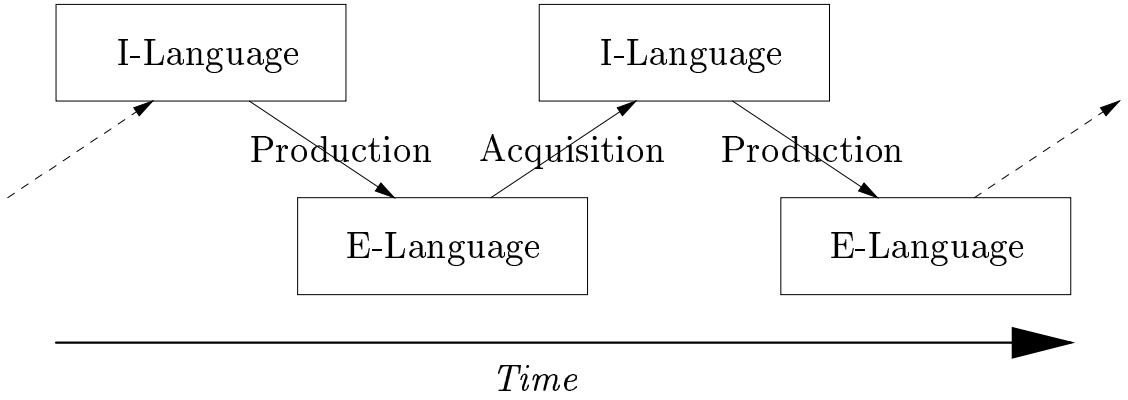


Figure 2.2: Language dynamic that results from repeated cycles of production and acquisition. (Adapted from Kirby, 2001, Fig. 5, p109.)

Rather than considering the dynamic of language evolution alone, Briscoe (1998) considers a co-evolutionary dynamic between language, and a Chomskyan style language acquisition device. Thus, in terms of Fig. 2.2, not only is the language being transmitted between generations via an acquisition device, but the properties of this device are also changing from generation to generation. Briscoe's results suggest that languages evolve subject to the learnability, parsability and expressivity requirements imposed by the linguistic environment, and that a truly co-evolutionary dynamic occurs despite the fact that changes in social phenomena (such as language) occur at a vastly different time-scale to genetic change.

Two micro-evolutionary models, in particular, have been major influences on this thesis. The first, by Batali, is known as the Negotiation Model (Batali, 1998, in press). The second, by Kirby is called the Iterated Learning Model (Kirby, 2000, 1999b, 2001). The primary distinction between these two models is the way in which individuals interact, that is, the structure of social interaction. Whereas

the population in Batali's simulations constitutes a single generation, in Kirby's simulations inter-generational dynamics are the driving force. Given the relevance of these two approaches to the work that is presented in later chapters, it is worthwhile to review them here in some detail.

The Negotiation Model

Batali's (1998) work represents the most significant connectionist account of language emergence in a population of simulated agents. Through a series of linguistic interactions a population of recurrent neural networks becomes adept at using a systematically structured language to communicate a set of concepts (representing meanings such as 'me happy' or 'you sick'). The concept representations were themselves compositional, being formed by the concatenation of a referent vector (one of ten vectors of four bits representing, for example, 'me') and a predicate vector (one of ten vectors of six bits representing, for example, 'happy'). Networks communicate by sending sequences of symbols (represented as four-bit vectors) to each other. On receiving a signal, an agent propagates the sequence of symbols through its weights, with the interpreted meaning being the final output of the network. Interestingly, the same network is used for both production and comprehension through the application of an algorithm that provides an approximation to the obverter procedure referred to earlier. The essential idea behind this approach is that to communicate some meaning M, an agent tries to produce an utterance U such that if it were to hear U itself, it would interpret it as meaning M.[11] The course of simulations is as follows.

1. Randomly select an agent from the population.
2. Randomly choose ten meanings and ten teachers from the population (one for each meaning).
3. Train the learner to correctly interpret the teachers' utterances with the back-propagation algorithm.
4. Return to step (1).

[11] If we consider an agent as a mapping from utterances to meanings $A : U \rightarrow M$, then an agent tries to produce utterances by the inverse of its own comprehension function A^{-1}. In Batali's work with recurrent neural networks, A^{-1} cannot be analytically determined from A so a local search is used to provide an approximation.

Interestingly, the populations are fixed; no agent enters or leaves. Rather, a population must reach a consensus on language conventions through a series of 'negotiations'. Batali reports that over time, populations do indeed converge on common languages. These language are expressive (all meanings are communicated uniquely, there are no homonyms or synonyms), consistent (all agents express meanings similarly) and comprehensible (agents understand each other's utterances). Moreover, the languages show some degree of systematic structure; 'predicate' and 'referent' roles can be identified in the agents' utterances corresponding to the predicate and referent values present in the meaning.

The interesting question that arises from the results of these simulations is why the population should converge on a systematic language. The space of possible languages is very large, and the vast majority are non-systematic in nature, so there appears to be an attractor for systematic structures in the dynamics of the negotiation process. On the issue of why this structure in particular emerges in the language, Batali postulates that the largest contributing factor to the observed results is the very nature of neural network representations. Neural networks of the type used by Batali have an implicit assumption of 'smoothness'. That is, they have a bias towards mapping similar input values to similar output values, the upshot of which is that it is simply more difficult to represent irregular mappings than regular ones. Consequently, the neural network agents in Batali's simulations inject a bias into the negotiation dynamic that predisposes the population towards finding systematic languages.

Neural networks are of course capable of representing complex mappings, so the potential is there for the population to converge on a language with an arbitrarily complex convention for mapping meanings to utterances. The systematicity of the observed languages is therefore an emergent property of the system. The dynamics of language interaction result in the emergence of a language which has the appearance of design.

These results demonstrate that significant properties of language, in this case compositionality, can be *emergent* properties from the dynamics of linguistic interaction and need not be pre-specified by an innate linguistic competence. There are, however, some potentially problematic issues in the design of the simulations. One of the more significant aspects of the negotiation model is the immortality of the population. Somewhat unrealistically, individuals never leave the population. The model fails to capture the interaction of an inexperienced, immature individual

learning from a mature one. As a consequence of this design, agents will eventually be exposed to every possible meaning — the agents are not required to *generalise*.[12] The other facet of the simulations that warrants closer attention is the dependency on an obverter-like procedure for producing utterances: what has come to be known as the 'inverse learning' requirement. At least in terms of the neural network implementation, it seems a somewhat anomalous mechanism for production.

The Iterated Learning Model

A different approach to realism in linguistic interaction is described by Kirby (1999b, 2000, 2001), who presents a compelling demonstration of the emergence of grammar in the absence of any phylogenetic adaptation.[13] A population of ten language users, modelled as context-free grammars, are arranged in a ring such that each individual has two neighbours. Individuals are capable of talking about simple meanings,[14] using strings produced from a restricted alphabet. Individuals are equipped with a learning mechanism, but the initial population has no grammar. That is, the initial population consists of a mechanism for *acquiring* language, but no language to acquire.

To bootstrap the system, Kirby introduces the notion of random invention: if an individual wants to communicate a particular meaning but has no way of expressing that meaning, it either says nothing or, with small probability, produces a random string. The course of a simulation runs as follows.

1. Replace a randomly chosen individual with a new individual.
2. Produce a corpus of training examples from the utterances produced by the new individual's neighbours.
3. The new individual induces an internal grammar based on this corpus.
4. Return to step (1).

[12] In later simulations, Batali (1998) allowed only 90 of the 100 meaning vectors to be used during the negotiation process. While agents were reasonably adept at generalising to the remaining 10 meanings after the negotiation process, the amount of generalisation required is small compared with human language acquisition.

[13] Unless otherwise noted, the description refers to the first simulations performed, though not the first published (Kirby, 2000).

[14] These meanings were represented as either <agent, action, patient> 3-tuples (Kirby, 2000), recursive 3-tuples (Kirby, 1999b), or 2-tuples with some probability distribution (Kirby, 2001).

At the start of a simulation run, the training corpora are typically small and contain examples that are more-or-less random. Gradually, the training corpora become larger as each individual's grammar becomes more expressive. After a period of time, individuals start to *regularise* their grammars in a compositional manner, using common substrings for common parts of a meaning. Eventually, the population comes to use a fully compositional language where every utterance can be broken into subcomponents, each representing a part of the meaning tuple.

Kirby deliberately chose the size of the training corpora so that it was highly unlikely that an individual would be exposed to the full set of *(meaning, utterance)* pairs. That is, the only way that an agent could acquire a complete grammar was to generalise from a limited subset of exemplars. Kirby hypothesises that it was this feature of the simulations — the 'learning bottleneck' — that caused the fundamental shift in the languages produced, from non-compositional to compositional.

If meanings and utterances are randomly associated, then there is no structure on which to base a generalisation mechanism. An unobserved association must therefore remain unknown. In contrast, with a systematic relationship between meanings and utterances, it is possible to generalise from a limited set of observed exemplars. This dichotomy, Kirby argues, introduces a 'glossogenetic' selection pressure for languages that can be expressed by a few general purpose rules and can be induced from a smaller set of examples. For these languages, it is not necessary to see every *(meaning, utterance)* pair. Instead, the general relationship between meanings and utterances can be derived from a suitably chosen subset of exemplars.

Not only do Kirby's results replicate Batali's major finding — that significant features of language can be emergent properties from the dynamics of language transmission — they also show that this phenomena can occur under more realistic assumptions about linguistic interactions. Furthermore, Kirby's later studies (Kirby, 2001) demonstrate how the same process responsible for the emergence of compositional language can (under some assumptions) be responsible for stable irregularity in a language.

Although there is no phylogenetic adaptation during the course of Kirby's simulations, the model incorporates phylogenetic adaptation implicitly in the design of the individuals' language learning mechanisms. That is, the starting point of the simulations is a population of individuals that are innately endowed with a particular learning mechanism. Although Kirby highlights the importance of languages themselves being systems that adapt to their human hosts, inherent in his choice of

learning algorithm is a strong form of language-specific learning bias. Kirby's language induction algorithm was originally developed specifically for computational linguistics so it is perhaps not surprising that the chosen algorithm is biased towards inducing language-like, compositional structures.

2.3 Dynamics of linguistic transmission

At this point, it seems appropriate to reconsider our motivating hypothesis (that language adapts to aid its own survival), in light of the results of the computational simulations presented above. The work surveyed above explores the issues associated with this hypothesis in greater detail. It suggests that language must be considered in terms of the dynamics of the linguistic interactions that allow it to propagate. Because language is always being propagated via production and acquisition (Fig. 2.2), it is inevitably being shaped by those two processes. Consequently, for a language to persist over time it must be capable of being transmitted unchanged through both processes. Such a language can be said to be a fixed-point of the dynamical system of linguistic propagation (assuming perfectly reliable transmission). An important issue is what happens to languages that are imperfectly transmitted, particularly whether the transmission dynamic has an attractor for particular forms of languages. The results on the obverter procedure (Oliphant and Batali, 1996; Batali, 1998) suggest that a necessary requirement for language to emerge is a particular relationship between the production and comprehension processes. This condition is by no means sufficient, and Kirby's (1999a) results emphasise the crucial role played by the learning algorithm which must recover an entire language from a limited set of examples that make it through the 'learning bottleneck'. While referring to the 'adaptation' of language is something of a biomorphism, the results presented above indicate that for a relatively stable language system to emerge, the dynamics of language transmission must be of a particular type: those that temper language into reliably transmissible forms.

2.4 Directions

We have discussed two frameworks for considering language — generative grammar and connectionism — highlighting the differences between the two, particularly on the subject of innate knowledge of language. We then introduced an alternative

hypothesis for explaining human infants' extraordinary linguistic talents: the adaptation of language. With computational modelling, this hypothesis may be applied to both generativist (Kirby, 2000) and connectionist (Batali, 1998) frameworks. This thesis focuses on the connectionist approach, concerning itself with probing the extent to which the adaptation of language can facilitate learning by general-purpose (connectionist) learning mechanisms. The work differs from and extends previous work by

- using a different semantic domain,
- considering language change when sender and receiver are computationally distinct,
- focusing on the potential generalisability of languages, and
- considering the impact of different styles of social interaction.

Chapter 3

Getting the Point Across

3.1 Modelling a language domain

The computational modelling of language origins requires significant infrastructure for simulations. The following issues must be addressed.

- *Who or what is communicating?* The design of the language agent defines its computational properties and plays a crucial role in determining the range of ways in which meanings can be mapped to messages (and vice-versa). The other factor in this design decision is the choice of learning algorithm which dictates how the agent incorporates the incomplete and inconsistent information available in the environment into its mechanisms for producing and understanding language. Previous work has considered neural networks (Batali, 1998; Cangelosi and Parisi, 1998), context-free grammar systems (Kirby, 2000), finite-state automata (Hashimoto and Ikegami, 1996) and other, more exotic systems such as robots (Steels, 1997a).

- *What are they trying to communicate?* Most models of communication assume that there is some underlying semantic domain on which language agents base their communication. Synthetic, abstract semantic information may be given directly to the agents, for example in the form of N-tuples (Kirby, 2000) or binary vectors (Batali, 1998). Alternatively, agents may inhabit some environment, the salient semantic features of which must be gleaned from the agents' interactions with the external world. Such environments may be real (Steels, 1997a) or simulated (Cangelosi, 2001).

- *How do they communicate?* Humans utilise a range of media for language, with spoken, written and signed forms being the most salient. It is apparent that (external) language can be described symbolically.[1] Consequently, most simulations assume some sort of symbolic communication channel (the only exception to this observation that the author is aware of is the work of Saunders and Pollack, 1996, who used a real value between 0 and 1 as the signal). Since simulations of language systems (as opposed to simpler signalling systems) utilise compositional linguistic structures, the communication channels of previous studies allow the transmission of a sequence of symbols. How the agents deal with this sequential aspect may have implications for the kinds of meaning/utterance relationships that can be formed.

- *How do they interact?* Language (and communication in general) is a *shared* task involving both a sender and a receiver. Moreover, within a population, linguistic interactions occur between many different pairs of senders and receivers. The issue of 'who talks to whom about what' may not immediately appear to be of much significance. However, as discussed in §2.2.1, the dynamics of linguistic interaction can be the driving force behind linguistic change. While some studies have considered language interactions as a logical outcome of an environment, most have introduced the notion of 'language games': a contrived series of interactions with the agents and task determined by the experimenter. Typically, the goal of the game is to communicate a meaning in a unidirectional manner (that is, without intermediate feedback from the receiver). The sender and receiver may be chosen randomly from the population, or the population may have a topology that limits interaction to within some neighbourhood. An additional design decision is the degree to which the receiver is given access to the intended meaning of the sender: in many simulations the receiver is given explicit access to the intended meaning as a basis for learning.

  A further aspect that warrants consideration is whether the population is static, or whether agents enter and leave the population over time. For non-static populations, the decision must be made on how and when agents are replaced. Rewarding successful communication may introduce genotypic selection, which former studies have often explicitly tried to avoid.

[1] The existence of written language demonstrates this point.

It is important to note that the goal of the design process is not necessarily to replicate as accurately as possible all of the features of the human environment. Not only is this task intractable, but it fails to be informative about which aspects of the environment are necessary and/or sufficient for language to emerge, and which are spurious. The design process is one of compromise. The need for an environment with an appropriate degree of complexity must be balanced by the need for a system that can be efficiently simulated and is amenable to detailed analysis. The simulations must be designed in such a way that the phenomenon of interest can be studied in isolation. To reiterate a point made in §2.2, the broader research goals are not to describe the conditions under which human languages evolved; it seems doubtful that such an outcome could ever be attained. Rather, the aim is to describe the necessary and sufficient (computational) properties that a system must exhibit to permit language-like communication systems.

The goal of this thesis is to explore the extent to which languages can adapt to be learned by general-purpose learners, and the conditions under which this adaptation might occur. The design of the simulation framework reflects this general goal. While each of the following chapters considers different models, the underlying simulation framework remains largely unchanged throughout the thesis. It is the general framework that will now be described, particularly with respect to the design issues raised above.

3.1.1 A simulation framework for studying the evolution of language

As noted in §2.1.2, the work in this thesis is presented within the connectionist paradigm. Hence, we have chosen to model language agents as neural networks. Particularly, we have chosen to use recurrent neural networks (RNNs) for their sequence-processing abilities and for their history of use in language domains (for example, Elman, 1991) and in former studies of language evolution (Batali, 1998). While there may be a history of applying RNNs in language domains, they are nevertheless general purpose learners; they were neither designed especially for language tasks, nor are they restricted to performing only those tasks. A variety of algorithms are available for training RNNs, the most well-known of which include ('vanilla') backpropagation, backpropagation through time (BPTT; Rumelhart et al., 1986) and real-time recurrent learning (RTRL; Williams and Zipser, 1989). Again, these

are general-purpose algorithms, based on generic gradient descent, and developed without regard to specific linguistic issues. While we expect the choice of learning algorithm to play a significant role, our issue is the adaptation of language to a learner, not one of finding the best learning algorithm. BPTT is the chosen algorithm since it is expected to be best suited to the task. (The on-line capabilities of RTRL are unnecessary, and BPTT may offer some advantages over standard backpropagation on temporal tasks.)

Whereas the choice of language agent was largely determined by a philosophical commitment to a particular paradigm (connectionism), the choice of semantic domain presents a wide variety of options. The structure of the semantic domain should, in some way, be reflected in the structure of its associated language. That is, utterances for related meanings should be related. This relationship is the basis for generalisation, a crucial element in studies of language. Creating a complex domain introduces the risk of making the learning task overly difficult.[2] In Cangelosi's (2001) work, the structure of the semantic domain is determined by the interactions of an agent with a simulated environment. However, for tractability and simplicity of analysis we shall consider an artificially created semantic domain. Whereas previous studies have used domains related to propositional logic (including Batali's 1998 neural network simulations), we will use a domain that has a qualitatively different structure. Quite simply, the semantic domain is the unit interval $[0, 1]$, a subset of the real numbers. Each 'meaning' or 'concept' is simply a point in this interval. While this domain is quite simple it nevertheless has some interesting properties. Foremost, is that this domain has an obvious distance metric. It is reasonable to talk about the distance between two meanings, giving a (continuously varying) measure of how well a concept is communicated. Furthermore, the domain is continuous; there are infinitely many different meanings so that the system may begin to differentiate, on finer-grained scales, between different concepts.

In keeping with tradition, utterances are modelled as sequences of (discrete) symbols. As with most connectionist modelling, symbols are represented with a one-hot encoding.[3] RNNs, which typically assume continuous output values, can generate such symbols in a variety of ways. The most common approach — winner-

[2] Of course, in the case of humans, the semantic domain has an amazing degree of complexity. However, as any modeller will attest, constructing a model that fails to exhibit interesting behaviour is a dishearteningly frequent occurrence. Creating a task with an appropriate degree of difficulty, so that the behaviour of the system is neither suppressed nor vacuous is the fundamental dilemma.

[3] For example, symbols a, b and c may be represented as vectors [1 0 0], [0 1 0] and [0 0 1] respectively.

takes-all — is to set the largest output activation to 1 and the remainder to 0. An utterance is thus a *sequence* of such transformed activations, produced sequentially by the sender, for example, $\langle[100], [010], [100], [001], [010]\rangle$ which might be denoted $\langle a, b, a, c, b\rangle$ or simply *abacb*.

The final major simulation design category, that of agent interaction, varies quite widely through the following chapters. Further discussion of this issue will take place independently in each chapter.

Given this sketch of the basic simulation framework, we are now in a position to consider the languages that might be useful in such a system. Clearly, we should not expect that human languages are the most suitable for this architecture. If human languages are subjected to any sorts of functional constraints, they are the functional constraints of their human users. That is, human languages are appropriate for communicating meanings as represented in the human mind, with human cognitive machinery, using human perception and action. There is little reason to expect that we could be so fortuitous as to have proposed a system with identical constraints to those of humans, so it should not be expected that there will necessarily be a significant similarity to the kinds of linguistic structures found in human languages. We might expect then, that the languages that do emerge from the system will reflect the underlying semantic domain, that is, the unit interval. Indeed, we find that the emergent languages exhibit properties similar to those of number systems. While the simulation design does not mirror the conditions under which human language emerged, it does allow the exploration of the *general principles* behind the emergence of structured communication systems.

3.1.2 On the role of bias in learning

As noted in §2.1, one of the major questions in linguistics concerns the reason that human languages assume certain forms. The set of observed human languages do not seem to vary limitlessly. Rather, they appear to be constrained by some universal principles. The adaptive-language hypothesis suggests that the properties of human learning mechanisms, to which languages must adapt, serve to constrain the range of viable languages. If the learning mechanisms that humans employ for language are not domain-specific, as some connectionists propose, then what makes one language more suitable than another? The conjecture that we consider in this thesis is that it is the (weak) inductive biases of (general-purpose) learners that act as the selection pressure for languages.

All learning algorithms include some form of bias. Quite simply, bias is the set of factors that determine the choice of hypothesis. The principle can be easily demonstrated by considering a one-dimensional non-linear regression problem. Given a finite number of samples from an unknown function, possibly with the addition of noise (ϵ_i), $g(x_1) + \epsilon_1, \ldots, g(x_n) + \epsilon_n$, the task is to recover the underlying function, $g(x)$. Standard practice is to postulate that g belongs to some parameterised class of functions, F.[4] That is, hypotheses of g are drawn from F. The choice of F introduces the first major source of bias, model bias, creating what is commonly known as the bias/variance dilemma (Geman et al., 1992). If F is chosen to be a class capable of representing arbitrary functions, it is said to have high variance. Conversely, classes that have limited representational capacity are said to have high bias. Choosing a class with high variance introduces the risk that the noise component of the samples may be incorrectly assumed to be part of the target function (known as overfitting). Choosing F to have strong bias risks excluding the target function (i.e., where $g \notin F$). This maxim is depicted graphically in Fig. 3.1. In the case of neural networks, the model bias is determined by the architecture of the network. In general, adding hidden units decreases the bias of a neural network (and reciprocally increases the variance).

Selection of a particular hypothesis space is not the only source of bias in this simple example. Once a class F has been chosen, the question remains as to which hypothesis to choose from F. Given the finite set of samples, there may be multiple hypotheses in F that are consistent with the data.[5] For example, in the case of our non-linear regression problem, if we choose F to be the class of pth order polynomials, and if there are no more than p samples, then there are infinitely many consistent hypotheses. Alternatively, if there are more than $p + 1$ points, there may be no consistent hypotheses. In this case the hypothesis needs to be selected on the basis of some measure of desirability, such as sum-squared error.[6] Optimising this metric with respect to the parameters of the model class may be a non-trivial exercise in itself, as is the case for neural networks. The manner in which a particular

[4] We shall avoid the issue of strictly non-parametric approaches.

[5] It is often the case that finding the consistent hypotheses in F is not a tractable task. This problem typically arises when F is chosen to be a general-purpose learner, such as a neural network (Blum and Rivest, 1992).

[6] For the simple regression problem here, the squared error for a particular example, x_i, is the squared difference between the actual value of the function, $g(x_i)$, and the hypothesised value of the function, $f(x_i)$. That is, $SSE(x_i) = (g(x_i) - f(x_i))^2$. The sum-squared error value is obtained by summing each of the squared errors over a set of examples, $x_0, \ldots, x_n$.

hypothesis is chosen introduces a search bias. Note that the matter of search bias is not unrelated to that of model bias since a larger hypothesis space will tend to have a greater number of consistent hypotheses from which to choose. (Of course, a consistent hypothesis may be suboptimal since the sample data may contain some degree of noise.) In the case of neural networks, the search bias is determined by the learning algorithm and its associated parameters (such as backpropagation) and the cost function.

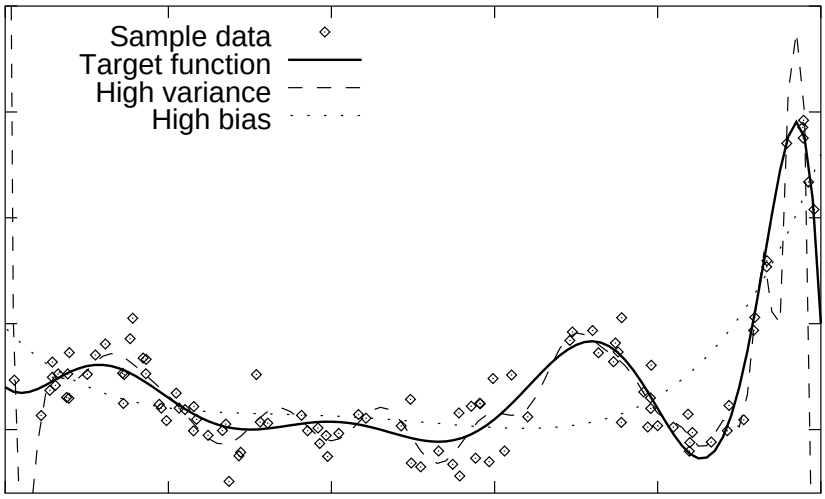


Figure 3.1: Non-linear regression demonstrating the role of model bias. A hypothesis space that is too large may cause overfitting. A hypothesis space that is too small may be incapable of representing the target function.

While there is much theory regarding the relationship between the complexity of the data and the complexity of the hypothesis space (most notably Valiant, 1984), it is difficult to put into practice. For many interesting choices of F, such as neural networks, the theoretical properties are difficult to establish. However, the general principles of the theory can be applied. They suggest that the bias of the hypothesis space, F, should be tailored to the known properties of the problem. Essentially, the bias of the algorithm should incorporate prior knowledge of the problem domain.

Returning to the issue of language learning, we see that generative grammar incorporates a learning mechanism with a high bias, tailored to a specific class of languages. This bias is primarily a model bias (the principles and parameters framework restricts the hypothesis space) that *a priori* constrains the range of languages that can be represented (i.e., the available hypotheses about which grammar is generating the observed utterances). The conjecture that is explored in this thesis is that languages can adapt to be learnable by mechanisms without such strong intrin-

sic bias by adapting to the search bias (or *inductive* bias) of their learners. What this conjecture implies for the human learner is that an infant's intuitions and generalisations about language would generally be correct because languages have evolved to exploit the kinds of assumptions human infants make.

3.2 Talking backwards

The issue that we take up in this chapter is what happens when the innate biases of sender and receiver are different. If languages adapt to be learnable by their users, is it possible for a learnable language to emerge when the most sympathetic language is different for sender and receiver? If one views sending and receiving as computationally distinct processes (albeit tightly coupled), one must also accept that the optimal language for each process may be different. Can a language mediate the different forces brought to bear upon it by the competing interests of its users? In this chapter we aim to demonstrate the principle that a language can adapt to accommodate opposing learning biases.

The paradigm for this work involves two recurrent neural networks, which try to communicate a meaning, represented by a point in the unit interval, $[0,1] \subset \mathbb{R}$. One network, the *encoder*, acts as the sender and is presented with points, $x_i \in [0,1]$. The encoder produces a sequence, $s_i \in \Sigma^*$, of symbols taken from an alphabet, $\Sigma = \{0,1\}$,[7] which is serially transmitted across a channel to a *decoder* network, acting as receiver. The decoder network receives the sequence as input, and outputs $y_i \in [0,1]$. If the communication is successful, then y_i should approximate x_i. The set of transmitted sequences, $S = \bigcup_i \{s_i\} \subseteq \Sigma^*$, forms a *language* or *code* for communicating the interval (see Fig. 3.2).

It is possible to accomplish this communication task using a *numeric* encoding, typically recognised as the "standard" binary representation (where the sequence of transmitted symbols corresponds to its base-two representation). There are two canonical ways in which such a sequence can be transmitted — either the most significant bits (MSB) of the message can be sent first, or the least significant bits (LSB) can be sent first.[8] Consequently, the initial simulations consider the sender

[7] The choice of 0 and 1 as symbols is arbitrary: 'a' and 'b' or 'ba' and 'di' would be equally appropriate.

[8] The question of the order in which bits should be sent is a notorious issue in computing that arises when two computers attempt to communicate over some serial communication medium (Cohen, 1981). Some manufacturers chose to send the least significant bit first (called Little-Endians

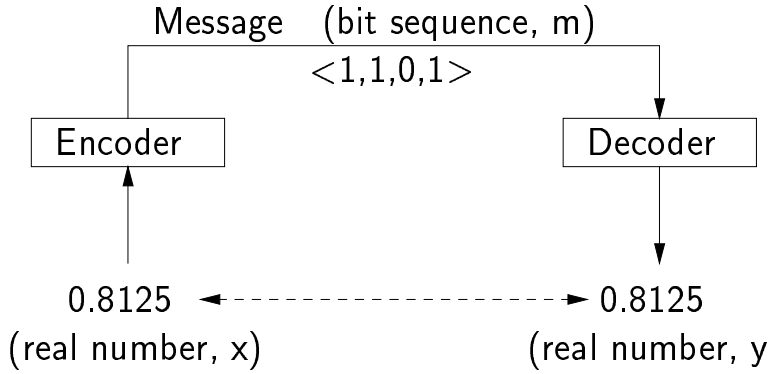


Figure 3.2: Getting the point across. Two recurrent networks are used as sender and receiver for a communication channel. The sender (encoder) is presented with a real number, $x \in [0, 1] \subset \mathbb{R}$ and outputs a sequence of symbols, $m \in \Sigma^*, \Sigma = \{0, 1\}$. This sequence of symbols is then used as input for the receiver (decoder), which outputs a value, $y \in [0, 1] \subset \mathbb{R}$, after all symbols in the sequence have been processed. If the communication is successful, then y approximates x. If a numeric encoding is used (see text) then the input value 0.8125 would be encoded as $\langle 1, 1, 0, 1 \rangle$, since $0.8125_{10} = 0.1101_2$. The decoder, upon receiving the sequence, would output 0.8125.

and receiver separately, trying to learn to produce and understand these two alternative languages (§3.3). These initial simulations suggest that, within this framework, the learning biases of the sender are in conflict with those of the receiver. Whereas the senders more easily learn the MSB-first languages, the receivers more easily learn the LSB-first (MSB-last) languages.

Given these initial results, we consider the combined sender-receiver system (where sender and receiver are at liberty to determine their own code) under two conditions (§3.4). In one simulation condition, the order of messages is reversed so that the biases of the sender and receiver are aligned. In the other condition, no reversal is performed. Analysis shows that the languages that emerge show a structural compromise between the competing biases of sender and receiver when no reversal is performed (§3.4.2).

In the final series of simulations, the sender is unrestricted in sending messages of varying length (§3.5), unlike the situation in earlier simulations where every message is of a fixed, predetermined length.[9] The simulations of §3.3 and §3.4 indicate that

in Cohen's Jonathon Swift-inspired account) whereas other chose to send the most significant bit first (Big-Endians) creating havoc when computers produced by different manufacturers wanted to communicate. While there are no meaningful parallels with the work presented here, the correspondence is an interesting one.

[9]For those familiar with Cohen's (1981) treatise on the Big-Endian versus Little-Endian debate in computing, by loose analogy this condition presumes no fixed word length.

decoders are able to exploit the existence of a fixed message length to enable them to more easily process MSB-first sequences. Removing this artifice strengthens the opposition between sender and receiver.

3.3 Study 1: Encoders, decoders and the numeric code

The first series of simulations investigates the ability of the individual encoders and decoders to perform their respective mappings in isolation. In total, four mappings are considered.

1. Encoding a real value as an MSB-first numeric sequence.
2. Encoding a real value as an LSB-first numeric sequence.
3. Decoding from an MSB-first numeric sequence to a real value.
4. Decoding from an LSB-first numeric sequence to a real value.

3.3.1 Encoders

The encoder is a first-order network, with recurrent connections from the output to the hidden units, as well as from the hidden units back to themselves (Fig. 3.3).[10] A single binary-threshold output unit codes `0` when off and `1` when on; a simpler alternative to a winner-takes-all approach, practicable when only two values are required. The networks are presented with a concept value at the first time-step, and an input value of 0 for subsequent time-steps. Concepts are chosen in accordance with a numeric binary encoding. The chosen concepts are those that can be encoded with exactly k bits (i.e., $\{n2^{-k}\}, 0 \leq n < 2^k$), and networks are given k time-steps in which to perform the encoding. These sets of concepts will be referred to as the k-bit values or *k-bit precision*. The sequence of outputs, taken from the network at each time-step is taken to be the encoding of the concept; the utterance associated with the given meaning.

[10] Essentially a combination Jordan/Elman network. This architecture is similar to that used by Christiansen and Chater (1999). The additional connection from the output to the hidden layer provides the hidden units with an explicit representation of what the network has output — the network "hears" what it "says". The additional weights also simplify the dynamics of the network.

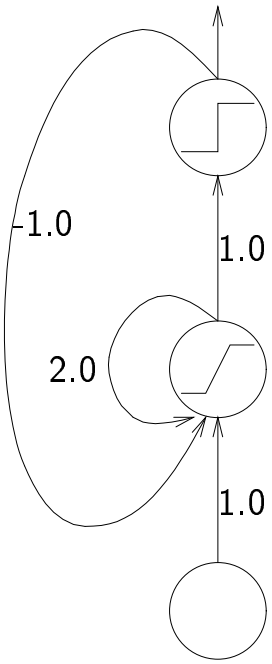


Figure 3.3: MSB First Encoder. A recurrent network that takes a real number in the unit interval $[0, 1]$ and encodes it to the numeric code, MSB-first. The hidden unit uses a linear threshold activation function that saturates at -1 and 1, and the output unit is a binary (0.5) threshold unit. The input value is presented at the first time-step only, subsequent inputs are 0. The network can encode values of arbitrary precision if allowed to produce a sufficiently long output sequence. In algorithmic terms, the recurrent weight of the hidden unit performs the equivalent of a "shift left" operation, and the negative weight from the output unit masks the "highest-order" bit. The activation of the hidden unit tracks the value that remains to be encoded.

Given this general architecture, it is relatively straightforward to hand-code a network with a single hidden unit to perform an encoding for a numeric MSB-first sequence. A linear-threshold activation function, as in Eqn. (3.1), is used for the hidden unit.

$$act(x) = \begin{cases} 1, \text{ if } x \geq 1 \\ -1, \text{ if } x \leq -1 \\ x, \text{ otherwise} \end{cases} \tag{3.1}$$

Such a network is shown in Fig. 3.3. By contrast, a network that performs the LSB-first encoding requires a large number of hidden units. For any value encoded with this scheme, the first output symbol is different to that of neighbouring values.[11]

[11]For example, with 4-bit precision the first symbol of the encoding for $\frac{3}{16}$ is 1, whereas for $\frac{2}{16}$

This encoding creates a fractal structure on the space which is difficult to process 'bottom-up' (i.e., from the fine-grained structure to the coarse structure). The fractal nature of the numeric encoding can be seen from a graphical depiction (Fig. 3.4).

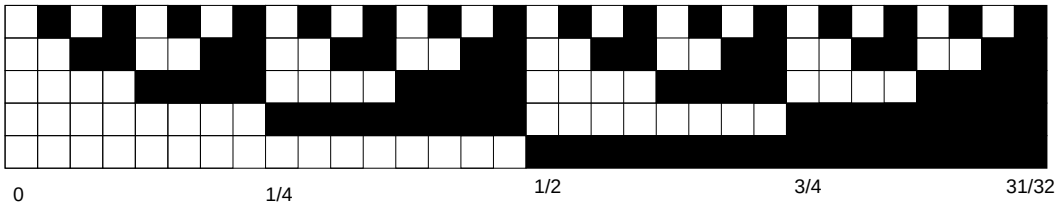


Figure 3.4: Representation of 5-bit numeric encoding. The range of concepts values varies along the x-axis, and the resulting code is shown against the y-axis. White areas represent the `0` symbol, and black areas the `1` symbol. The MSB encoding may be read off bottom-to-top, the LSB encoding top-to-bottom. For example, $\frac{1}{2}$ is encoded as $\langle \mathtt{1}, \mathtt{0}, \mathtt{0}, \mathtt{0}, \mathtt{0} \rangle$ (MSB-first) and $\langle \mathtt{0}, \mathtt{0}, \mathtt{0}, \mathtt{0}, \mathtt{1} \rangle$ (LSB-first).

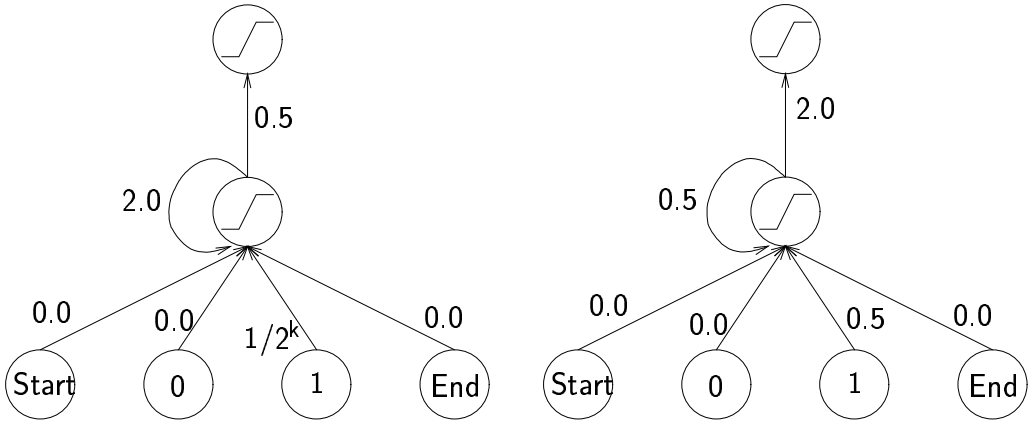
3.3.2 Decoders

Simple recurrent networks (Elman, 1990) were used for the decoders. The task for the decoders was the inverse of the encoder's task with minor variations. Given a sequence of symbols, the decoder was required to produce the corresponding concept. The difference from the encoder's task was that each sequence presented to a decoder was enclosed by start-of-sequence and end-of-sequence markers. For example, an encoding of $\langle \mathtt{0}, \mathtt{1}, \mathtt{0}, \mathtt{1} \rangle$ would be presented to the decoder as $\langle \mathtt{\#}, \mathtt{0}, \mathtt{1}, \mathtt{0}, \mathtt{1}, \mathtt{\$} \rangle$, where `#` and `$` are start and end symbols respectively. These additional symbols become important later when we consider variable-length encodings (§3.5), but did not appear to greatly influence the results presented in this section. To accommodate this change in the symbol set, the symbols sent by the encoder were recoded from their original single binary values to a (four dimensional) one-hot encoding. The encoders were not required to produce these symbols as it added unnecessary complexity.

Unlike the encoder, the decoder is capable of decoding either MSB- or LSB-first, albeit with some important asymmetries. Fig. 3.5 shows hand-coded simple recurrent networks that decode (a) MSB-first and (b) LSB-first. Although an LSB-first decoder is able to decode sequences of varying lengths with only a single hidden unit, an MSB-first decoder with the same architecture can only decode strings of

and $\frac{4}{16}$ the first symbol is `0`.

a known length. That is, for the MSB-first decoder the solution for k-bit precision does not generalise to $(k+1)$-bit precision, whereas for the LSB-first decoder it does. With three hidden units (whose activations are initialised to zero before presentation of a string) it is possible for the MSB-first decoder to process strings of arbitrary length (Fig. 3.6). However, we have never observed this solution as a result of learning, indicating that it is difficult to find for the learning algorithms we have used.



(a) MSB-first

(b) LSB-first

Figure 3.5: Simple recurrent networks that decode numeric sequences (a) MSB-first and (b) LSB-first. The input to the network is wrapped with start and end markers. After presentation of the end marker, the output unit activation is the value of the concept corresponding to the input sequence. Linear threshold activation functions (Eqn. (3.1)) are used for hidden and output units on both networks. Whereas the LSB-first decoder (b) is able to decode sequences of arbitrary length, the MSB-first decoder (a) can only decode sequences of known length, k, with one of its weights dependent on this value. In both cases, activations of the hidden units are set to zero before each string is processed. The MSB network effectively performs the operation: $\texttt{output}_{i+1} = \texttt{output}_i \times 2 + \texttt{input}_{i+1} \times 2^{-k}$, where k is the length of the sequence. The LSB decoder works in the opposite direction, computing $\texttt{output}_{i+1} = (\texttt{output}_i + \texttt{input}_{i+1})/2$.

3.3.3 Learning a fixed language

Although solutions could be hand-coded for the static language mappings (at least in three of four cases), it was unknown whether a solution could be learned. A series of simulations was designed to test whether the MSB-first or LSB-first codes could

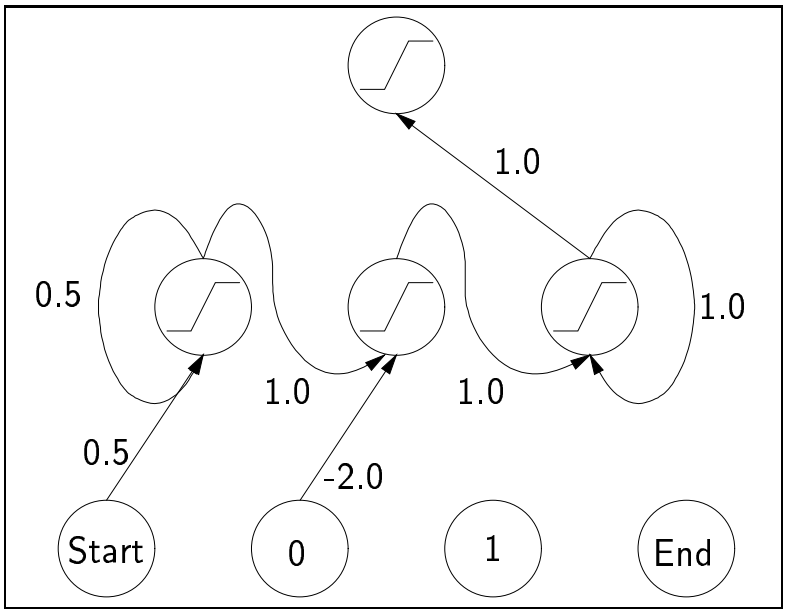


Figure 3.6: A recurrent network that decodes numeric sequences of arbitrary length MSB-first. Weights not shown have value 0. The network performs the MSB decoding by maintaining the value 2^{-i} on the first hidden unit and propagating this value to the third hidden unit. The weight from the `0` input to the second hidden unit acts as a gate on this propagation, so that the activation of hidden unit three is unchanged on `0` input. In this solutions the hidden units delay the propagation of the intermediate values and the `$` symbol is necessary for the solution to arrive at the output unit at the right time. As for the former decoders, activations of the hidden units are reset to zero before each string is processed.

be learned by either encoders or decoders. Given the relative ease in hand-coding solutions for the four mappings, we expected that weights for LSB-first encoders and MSB-first decoders would be harder to find than for MSB-first encoders and LSB-first decoders. In place of a gradient descent learning algorithm, we use a simple (1+1)-Evolutionary Strategy (Bäck and Schwefel, 1993) to optimise the weights.[12] This algorithm has much in common with a simple hill-climber — both ensure that any accepted solution is better than the current-best solution — and for reasons of ideological similarity, the algorithm used in these simulations will be referred to as a hill-climbing algorithm.

For this algorithm, a "champion" network was created with initially random weights, distributed uniformly between -0.1 and 0.1. A single mutant was then generated by randomly perturbing the weights of the champion according to a $\mathcal{N}(0, \sigma)$

[12]This technique has proven effective at finding recurrent networks that process reasonably complex languages (Tonkes et al., 1998; Chalup and Blair, 1999, for example,).

normal distribution. If the mutant was better than the champion at encoding or decoding the language, then the mutant became the new champion. Another mutant was then generated from the champion. The performance of each network was assessed by assuming an ideal numeric encoder or decoder counterpart. The fitness of an agent was thus the sum-squared error between encoder input and decoder output. It is this fitness value that determined whether or not the mutant network was superior to the champion network.

Since the encoders use a binary threshold output, a small change in the weights may have no effect on the output of the networks (i.e., the activation landscape consists of a series of steps, rather than being a smoothly varying surface). Consequently, the standard deviation (σ) with which mutants were generated was modulated throughout the course of the simulations. Whenever the mutant and champion encoded the input-set equally well, σ was increased by 0.1%, to broaden the search. In the case of equally-good encodings, the mutant is declared the winner so that the space around a particular solution is better explored. Furthermore, whenever a mutant lost to the champion, σ was mutated with 1% Gaussian noise, with an upper limit of 0.1. No change was made to σ when the mutant won. The same scheme was employed for simulations performed with the decoders, although in this case it was improbable for mutant and champion to perform equally well, since a decoder's continuous output means that it does not share the same type of discretised activation landscape.

The concept values chosen to be communicated were selected by taking a "staged learning" approach that has previously been used for language learning tasks (for example, Tonkes et al., 1998). For the encoders, initially only two values, 0 and $\frac{1}{2}$, were presented with the target single-symbol encodings being $\langle \mathtt{0} \rangle$ and $\langle \mathtt{1} \rangle$ respectively for both the MSB- and LSB-first encodings. Once an encoder was able to perform this mapping, $0, \frac{1}{4}, \frac{1}{2}$ and $\frac{3}{4}$ were encoded into 2-symbol sequences: $\langle \mathtt{0}, \mathtt{0} \rangle, \langle \mathtt{0}, \mathtt{1} \rangle, \langle \mathtt{1}, \mathtt{0} \rangle$ and $\langle \mathtt{1}, \mathtt{1} \rangle$ respectively in the MSB-first case, and $\langle \mathtt{0}, \mathtt{0} \rangle, \langle \mathtt{1}, \mathtt{0} \rangle, \langle \mathtt{0}, \mathtt{1} \rangle$ and $\langle \mathtt{1}, \mathtt{1} \rangle$ respectively in the LSB-first case. In general, once 2^k values could be successfully communicated using k-symbol sequences, encoders were given 2^{k+1} values to encode into $(k+1)$-symbol sequences. The activations of the networks were reset to 0 before each value was encoded.

Similarly, decoders were initially presented with the strings of length one, bounded by start and end markers, $\langle \mathtt{\#}, \mathtt{0}, \mathtt{\$} \rangle$ and $\langle \mathtt{\#}, \mathtt{1}, \mathtt{\$} \rangle$. When decoders could decode these strings to an appropriate degree of accuracy, the length (and number) of strings was

increased. Decoders were judged to be "correct" when the decoded value was closer to the relevant encoded value than any other value in the input set (i.e., within $\frac{1}{2^{k+1}}$ of the desired value for k-bit precision). Again, activations of the decoder were reset to 0 before presentation of each sequence.

Simulations were run for a maximum of 100K generations, or until all 32 5-bit values could be communicated accurately. Simulations were performed using networks with 1, 2, 3 and 5 hidden units. Fifty encoders and fifty decoders were evolved in each condition for both LSB- and MSB-first encodings.

3.3.4 Encoder and decoders: Results

As expected, the results of these simulations (shown in Table 3.1) indicate that there is an asymmetry in the sequence order preferred by the encoders and decoders. Whereas the encoders were far more successful on MSB-first sequences, the decoders were more successful at finding solutions to LSB-first sequences. For the encoders in the LSB case, no network was ever able to encode more than 2-bit values, whereas in the MSB case, networks of all sizes were able to encode 5-bit values, the most successful of which (in terms of the number reaching the maximum precision) were the networks with 2 hidden units.

Table 3.1: Performance of encoders and decoders on MSB- and LSB-first tasks. Shown are the number of networks, from the fifty trials, that were able to encode or decode 5-bit values within 100K generations, and the average precision finally obtained.

	Hidden Units	Number at 5-bit Precision (out of 50)		Average Precision (standard deviation)	
		MSB First	LSB First	MSB First	LSB First
Encoders	1	11	0	2.00 (1.28)	0.98 (0.14)
	2	18	0	2.84 (1.75)	1.14 (0.35)
	3	11	0	3.00 (1.29)	1.28 (0.45)
	5	7	0	3.20 (1.03)	1.56 (0.50)
Decoders	1	0	19	1.22 (0.55)	2.42 (2.06)
	2	0	26	1.62 (0.53)	3.28 (1.88)
	3	0	30	2.02 (0.59)	4.14 (1.29)
	5	0	22	2.39 (0.85)	3.82 (1.40)

Of the decoders presented with strings LSB-first, 30 of 50 were able to perform the task for 5-symbol input sequences. The best performance observed was with

a decoder with three hidden units. No MSB-first decoders were successful on 5-symbol sequences, although with 5 hidden units, one network out of the 50 was able to process 4-symbol sequences and 27 could process 3-symbol sequences. The asymmetry of these results suggests that the encoder, combined with the hill-climbing algorithm, has a bias towards learning MSB-first languages, whereas the decoder and hill-climber combination has a bias towards learning LSB-first languages.

3.3.5 Backpropagation through time

Hill-climbing is an atypical algorithm for neural network learning. It is an 'undirected' search method and can be quite slow. It thus makes sense to employ a more conventional learning algorithm, in this case BPTT. In the combined system there is no sensible way to train the encoder with this algorithm since the language is not fixed and there are consequently no target outputs. However, if we assume that the decoder "understands" what the encoder is communicating (i.e., the decoder has as a target value the input to the encoder; a common assumption in many studies of language learning) then the decoder may be trained towards a target in the typical manner.

Backpropagation works best with an activation function with a non-zero gradient everywhere, so we replaced the linear threshold activation function used for the hill-climbers (which has zero gradient when the summed input is not between -1 and 1) with Eqn. (3.2) which is linear in the interval $[-1, 1]$ and has a non-zero gradient everywhere.

$$act(x) = \begin{cases} 2.0 - \frac{1}{x}, \text{ if } x > \quad 1.0 \\ -2.0 - \frac{1}{x}, \text{ if } x < -1.0 \\ x, \text{ otherwise} \end{cases} \tag{3.2}$$

For training, networks were unfolded for up to 5 time-steps, but no further than the start of a message. Networks were only given a target value on presentation of the end-of-sequence symbol, \$; no errors were propagated as a result of intermediate outputs.

Simulations were performed for both MSB- and LSB-first encodings, again using a staged learning approach, in much the same manner as the simulations in §3.3.3. The networks' performances were tested after each epoch and precision was incremented accordingly. Decoders were trained for a maximum of 10K epochs with a learning rate of 0.01 and a momentum value of 0.9.

A similar set of results was obtained for the decoders trained with BPTT as was obtained using the hill-climbing algorithm. The success of the decoders in each condition is documented in Table 3.2. Decoders of all sizes learned to decode values of 5-bit precision when presented with LSB-first sequences. No decoders attained this level of precision when presented with MSB-first sequences, although some networks with more than a single hidden unit could decode 4-bit values.

Table 3.2: Performance of decoders trained with BPTT on both MSB- and LSB-first tasks. Shown are the number of networks, from the fifty trials, that were able to decode 5-bit values within 10K epochs, and the average precision finally obtained (*cf* Table 3.1).

Hidden Units	Number at 5-bit Precision (out of 50)		Average Precision (standard deviation)	
	MSB First	LSB First	MSB First	LSB First
1	0	24	1.08 (0.40)	2.92 (2.02)
2	0	35	1.80 (1.40)	3.88 (1.90)
3	0	20	2.06 (1.71)	2.64 (2.17)
5	0	7	0.86 (1.16)	1.36 (1.76)

Again, the performance of the LSB-first decoders exceeded that of the MSB-first decoders. The performance of the larger networks was disappointing, with many failing to decode values of even 1-bit precision and many others failing at 2-bit precision. The BPTT-trained decoders were often able to find good solutions more quickly than the hill-climbers, in terms of both the number of weight updates and simulation time.[13] Indeed, in the LSB-first condition, many networks were able to decode 5-bit values after as few as 200 epochs.

These results provide further indication that the decoders have a bias for LSB-first languages over MSB-first languages. Even with a different learning algorithm, with an inevitably different search bias, LSB-first languages were more readily learned. This result suggests that either the learning biases of BPTT and hill-climbing are acting similarly (quite plausible given that both are trying to minimise squared error), or that the architectural bias of the network is playing a significant role.

[13]BPTT has far greater computational complexity per weight update than hill-climbing.

3.4 Study 2: The combined system — forwards and reversed

Having established that both encoders and decoders are capable of learning a predetermined code in isolation (for at least some conditions), we turn to the question of whether the combined system is able to develop its own code (as originally described in Fig. 3.2). Rather than fixing the language and finding adroit encoders and decoders, we allow the language to vary with the qualification that communication should be successful (that is, the decoder's output should approximate the encoder's input). Note that unlike Batali's (1998) simulations, sender and receiver are independent. There is no explicit mechanism through which a change to the encoder causes a change in the decoder; the decoder must actively learn when the language produced by the encoder changes.

The results of study 1 (§3.3) suggest that the encoder and decoder have quite different biases in terms of the codes that are easier to learn. More specifically, we are led to the conjecture that if a code is among those acceptable for the encoder, then the *reverse* of that code is likely to be acceptable for the decoder. To test this conjecture we conducted experiments of the combined system under two different conditions: the *forwards* condition, in which the symbols produced by the encoder are passed on in the same order to the decoder, and the *reversed* condition, in which the symbols produced by the encoder are effectively buffered on a stack, and then presented to the decoder in the reverse order to which they were produced by the encoder.

As noted in §3.1, the type of interactions that occur between the agents of a community can play a significant role in language emergence. The simulations presented in this section employ a very simple dynamic, using only a single encoder and a single decoder. Preliminary simulations attempted to use the hill-climbing algorithm simultaneously on both the encoder and decoder. These simulations proved unsuccessful, with both the forwards and reversed systems failing to produce expressive languages. Consequently, an asymmetric approach was taken, applying BPTT to the decoder, and reserving hill-climbing for the encoder. Encoders were evaluated by training a decoder (with random initial weights) using BPTT on the languages they produced. This approach encourages the output of the encoder towards a code that is learnable by the decoder.

Random mutations to encoders often result in uninteresting languages, where

relatively few different utterances are used to communicate the interval (i.e., the set of utterances is highly homonymous). Since learnability by BPTT is an expensive function to evaluate, computational tractability required a simple precursory examination to eliminate poor encoders. Encoders were thus screened on the basis of their *variability* (the number of different encodings they produce for the set of inputs). If the mutant failed to demonstrate greater variability than the champion, it was discarded without training a decoder. Thus, there is an artificially introduced selective pressure for variability in the encoders (they are required to "babble").[14] Note however, that encoders with more variable codes will still only survive if the code they produce is learnable by the decoder.

To summarise, languages are evolved by hill-climbing in the weights of the encoder. Mutated encoders are subjected to a two-stage evaluation function, comprising (a) the number of unique strings produced for the input set; and (b) the sum-squared error of a random decoder, trained on the output of the encoder. Again we apply the principle of staged learning, initially using only $1-$bit precision, incrementing by 1 each time an encoder and decoder can successfully communicate all concepts, to a maximum of $5-$bit precision. The process is summarised in Fig. 3.7.

The interactions in this model introduce an asymmetry between encoder and decoder. The language of the system is determined by the encoder, which changes slowly over time, and does not learn in the orthodox sense. Conversely, with each small change in the encoder, an entirely new decoder is trained. Nevertheless, the biases found in §3.3 remain. Candidate encoders should still tend to produce MSB-first languages, and each decoder should still find LSB-first languages easier to learn. The conflicting biases express themselves in the encoder; LSB-first languages should (in the forwards system) lead to higher fitness, but it is difficult to find weights for an encoder to generate these languages.

One further aspect of the simulations warrants attention. In study 1, only the minimum number of symbols were sent: when 2^k values were being encoded, k symbols were sent. For the combined system this condition was relaxed to permit more than the strict minimum number of symbols to be sent, thus allowing codes that achieve less than optimal efficiency. For the reversed systems, values of $k-$bit precision were encoded into $k+2$ symbols, and for the forwards systems, values of $k-$bit precision were encoded into $2k$ symbols. Greater bandwidth was given to the

[14] When this condition was weakened to allow mutants to be of equal variability but producing a different encoding, simulations were typically unsuccessful within the limited time afforded them.

1. Set $k = 1$.
2. Create an initial champion encoder with random weights distributed uniformly between -0.1 and 0.1.
3. Train a random decoder with BPTT on the output of the champion encoder for n epochs. In each epoch, present all $k-$bit concepts. After training, compute the squared error between champion encoder input and decoder output, summed across all 2^k concepts. Assign this error to the champion encoder.
4. Create a mutant encoder.
   (a) Create a mutant encoder by perturbing the weights of the champion with $\mathcal{N}(0, \sigma)$ Gaussian noise.
   (b) Present each of the 2^k meanings to the mutant encoder. Calculate the number of unique strings produced.
   (c) If the mutant encoder produces more unique strings than the champion encoder (or if the mutant encoder produces 2^k unique strings), proceed to step (5). Otherwise, return to step (4a).
5. Repeat step (3), but for the mutant encoder.
6. If the error assigned to the mutant encoder is less than that assigned to the champion encoder, make the mutant encoder the new champion. Furthermore, if the mutants correctly communicate all values, increment k. (A concept is correctly communicated if the absolute difference between encoder input and decoder output is less that 2^{k+1}.)
7. Return to step 2.

Figure 3.7: Algorithm for combined encoder/decoder system.

forwards system since, unlike the reversed system, we do not expect it to develop a code as compact as the numeric code.

Fifty systems were evolved in both the forwards and reversed conditions, for a maximum of 100 generations[15] or until all 5-bit values could be successfully communicated. Decoders were trained for $k \times 750$ epochs, k being the precision of the communicated values. Decoders were trained under the same conditions as those in §3.3.5: BPTT with a learning rate of 0.01 and a momentum of 0.9. Both encoders

[15] One generation being the selection of a mutant encoder and the subsequent training of a decoder.

and decoders had 2 hidden units.

3.4.1 Forwards and reversed: Results

Simulations produced systems capable of communicating values of varying levels of precision, show in Table 3.3.[16] The forwards systems, sending $2k$ symbols, attained an average precision of 3.18 bits, whereas the reversed systems, sending $k+2$ symbols attained an average precision of 4.36 bits. The evolution of the system is clearly more successful when the communicated sequence is reversed. This result is not unexpected since the natural biases of the encoders and decoders push the system towards a solution in the reversed case, whereas the path to a successful solution is less clear in the forwards case.

Table 3.3: Performance of learners grouped by the level of precision obtained by systems in both the forwards and reversed conditions.

Final Precision Reached	Forwards ($2k$)	Reversed ($k+2$)
3	41	1
4	9	30
5	0	19
Total	50	50

3.4.2 Analysis of codes

In this section we analyse what, if any, were the differences between the codes produced in the forwards systems and reversed systems. This analysis is performed by (a) visual inspection of the emergent codes, and (b) measurement of how quickly the information that the code provides to the decoder increases with successive symbols.

In all cases, systems trained with the *reversed* channel produced codes similar in nature to the MSB-first numeric code of section 2. One code produced by an encoder with a reversed channel, for 5-bit precision, is shown graphically in Fig. 3.8 (*cf* Fig. 3.4). Some similarities are apparent between the evolved code and the numeric code. The similarities between the evolved code and the numeric code can

[16]Note that whereas Table 3.1 and Table 3.2 show how many networks attained 5-bit precision for varying numbers of hidden units, this table and Table 3.5 document the final precision attained by each system, having 2 hidden units for both encoder and decoder.

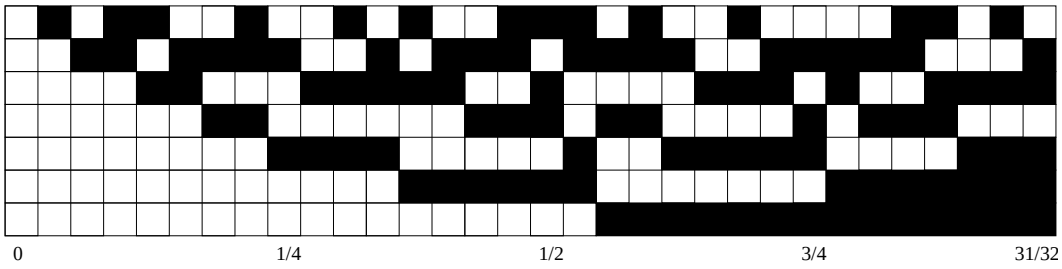


Figure 3.8: Graphical depiction of a code for communicating 5-bit values produced by a system from the *reversed* condition. The first output symbol from the encoder (and consequently the last input symbol to the decoder) is shown as the bottom row. Note the similarities to the numeric binary code shown in Fig. 3.4.

be seen by examining the rate at which each bit changes across the concept space. Over the range of inputs, the first symbol (shown as the bottom row of Fig. 3.8) alternates only once: the first symbol for encoding the smallest seven inputs is a `0`, and for the largest nine inputs a `1`. For the second symbol four such sequences are apparent, with an increasing number for subsequent symbols. The code shows a clear significance from first bit to last, as found in the numeric codes. Moreover, the direction of significance (MSB-first) is as anticipated.

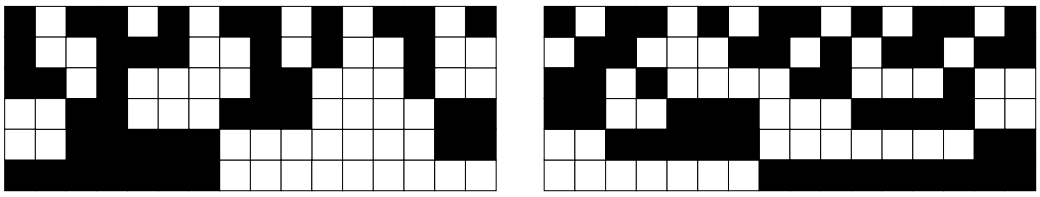
Many of the evolved codes did not share such obvious similarities with the numeric code as that in Fig. 3.8. Table 3.4.2 and Fig. 3.9 shows a typical code communicated by a reversed system for 4-bit values. On initial inspection, this code appears to be unrelated to a numeric code. However, interchanging symbols in the first, third and fifth positions in the sequence (i.e., replacing `0` with `1` and vice-versa) place the messages in the same numeric ordering as the inputs. This type of transformation was common to many codes but was by no means universal, others already being in numeric order (or reverse numeric order). The phenomenon may be attributed to negative recurrent weights that invert the significance of alternate symbols.

To provide a more direct comparison between the codes from the forwards and reversed systems, an additional 50 reversed systems were evolved allowing $2k$ symbols to be sent by the encoder, as used in the forwards case. Of these 50 systems 46 attained at least precision 4, the maximum attained by the forwards systems. Additional systems were also trained in the forwards condition, to bring the total number of forwards systems at 4-bit precision to 14. These 14 forwards and 46 reversed 4-bit codes were further analysed to study the effect of the differing biases on the form of the evolved code.

Table 3.4: Language for a *reversed* system at 4-bit precision. Interchanging symbols in alternate positions in the message (symbols 1, 3 and 5), shown in the third column, transforms the code into a numeric order. The symbol-swapping behaviour is a consequence of having negative recurrent weights that oscillate the interpretation of successive symbols. These codes are depicted in Fig. 3.9.

Input	Message	Alternately Negated	Output
0.0000	100111	001101	0.0029
0.0625	100100	001110	0.0420
0.1250	111001	010011	0.1211
0.1875	111111	010101	0.1943
0.2500	110010	011000	0.2738
0.3125	110011	011001	0.3136
0.3750	110000	011010	0.3608
0.4375	001001	100011	0.4424
0.5000	001111	100101	0.4945
0.5625	001100	100110	0.5412
0.6250	000011	101001	0.6105
0.6875	000000	101010	0.6577
0.7500	000001	101011	0.7272
0.8125	000111	101101	0.8002
0.8750	011000	110010	0.8488
0.9375	011001	110011	0.9184

A cursory inspection of the two sets of codes did not suggest any obvious structural differences, both sets of codes resembling the numeric code to some extent. What we expected to observe in the forwards codes was a bias to balance the information more evenly throughout a sequence, to cater for the biases of both the



(a) normal (b) alternately negated

Figure 3.9: Graphical depiction of codes listed in Table 3.4.2. (a) The code as produced by the encoder. (b) The code with every other symbol negated (`0` and `1` interchanged).

encoder and decoder. To find how information was distributed throughout the codes we considered the optimal squared-error (OSE) that a decoder could possibly obtain if it only saw the first n bits of the code, for $0 \leq n \leq 2k$. If the first n bits of each message are sufficient to identify it uniquely from the messages sent for other meanings, then each point can be precisely determined and the OSE of the code is 0. If two or more messages share the same initial n bits, then the optimal decoder outputs their average value, and the OSE can be calculated accordingly.

For an MSB-first numeric code, OSE drops very quickly as n increases since the most significant bits are at the beginning of the sequence. Conversely, for an LSB-first numeric encoding, OSE decreases slowly for the first values, then very rapidly for later ones. Fig. 3.10 shows these curves for numeric 4-bit MSB-first and LSB-first numeric encodings. Note that while the curves for the numeric code are symmetric, this observation does not necessarily follow for the evolved codes which are longer than optimal.[17]

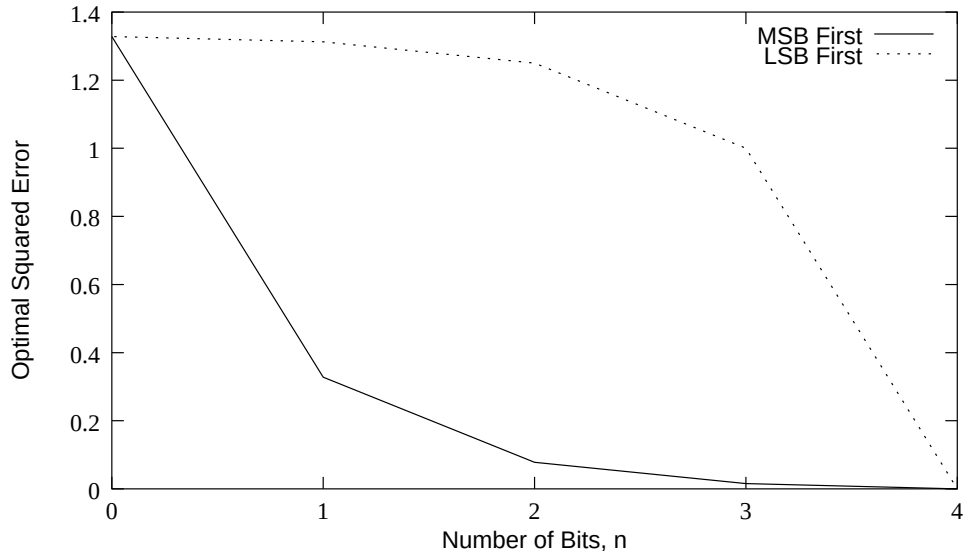


Figure 3.10: Least squared error obtainable by an optimal decoder after seeing only the first n bits of MSB-first and LSB-first numeric codes.

For the evolved codes, two scenarios are considered: the OSE of the code when read from the first bit sent by the encoder to the last bit sent (i.e., from the perspective of the encoder); and from the last bit sent by the encoder to the first bit sent (as would be seen by a decoder in the reversed condition). These statistics can be used to indicate where the 'effective information' is positioned in the code. The

[17] Consider a code formed by the concatenation of an MSB-first code and an LSB-first code, resulting in perfect information in the first half of the code when viewed in either direction.

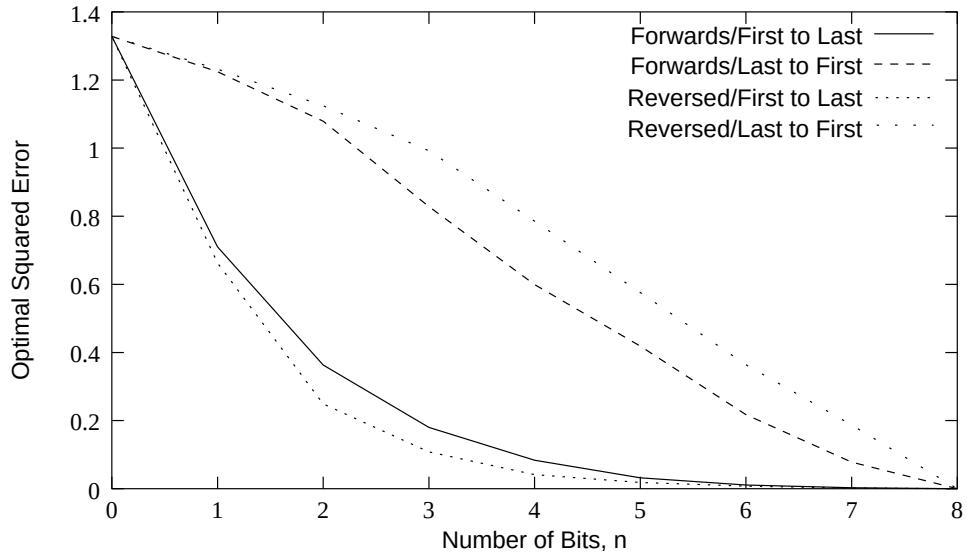


Figure 3.11: Least squared error obtainable by an optimal decoder after seeing the first n bits of codes developed by the forwards and reversed systems when read both first-to-last and last-to-first. The lower two curves represent languages that are presented first-to-last (i.e., as they normally would be in the forwards condition) while the upper two curves represent languages presented last-to-first (i.e., as they normally would be presented in the reversed condition). The two inner curves are from the languages produced from forwards systems and the outer two curves are from language evolved in the reversed condition.

resulting pair of curves is effectively the same as for the graph in Fig. 3.10 since an LSB code is simply an MSB code viewed in reverse. Fig. 3.11 shows the average OSE for both the forwards and reversed systems, when viewed both first-to-last and last-to-first.

Comparing the pairs of first-to-last/last-to-first curves in Fig. 3.11 with those of Fig. 3.10 reveals some of the differences between the codes of the forwards and reversed systems. The MSB-first numeric code has the strongest possible ordering of significance from first bit to last, and the area between the MSB-first and LSB-first curves is consequently large. In Fig. 3.11 the area between the curves of the reversed systems is larger than that for the forwards systems, suggesting that the reversed systems tend to develop codes with stronger ordering of significance than the forwards systems, as predicted. Calculating the area between the curves for each code in both the forwards and reversed systems, and comparing the two groups using a Mann-Whitney U test[18], reveals a statistically significant difference in the

[18]The Mann-Whitney U test is a non-parametric version of the t test.

two populations ($p < 0.05$). The graph demonstrates that the effect of the opposing biases in the forwards system is to distribute information more evenly throughout the sequence.

The differences between the codes of the forward and reversed systems, while significant, are not extreme. The similarities may be attributable to the encoder having a greater bias than the decoder. When the language was fixed in study 1 (§3.3), the MSB decoders were far more successful than the LSB encoders, possibly because the networks took an approach similar to the MSB-first decoder in Fig. 3.5. Preventing this solution then, may make the strength of biases between encoders and decoders more consistent, thus causing a greater distinction between codes evolved in the forwards and reversed conditions.

3.5 Study 3: Variable length codes

In the simulations presented thus far, the encoders have generated an externally specified number of symbols (either $k + 2$ or $2k$). In the following series of simulations, encoder networks have an additional output unit that can be used to control the length of sequence produced. So long as this additional output remains on, symbols are sent in the same manner as the previous simulations. When this output unit turns off, or some maximum number of symbols is reached, an "end-of-sequence" (EOS) symbol is added to the message, and communication ceases. The additional output unit may be considered as a "push-to-talk" unit.[19] This change allows encoders to send the same sequences when the number of values to be communicated increases.[20] It also makes success less likely for the type of MSB-first decoder in Fig. 3.5 which can only process sequences of known length.

Simulations were performed in much the same manner as those in the previous section, with some minor changes regarding EOS. In both the previous and following series of simulations, encoders were selected on their ability to produce a wider variety of codes. In the following simulations, once an encoder attains maximum variability, subsequent encoders are generated to maintain variability in the codes, but also to increase the number of EOS symbols produced (i.e., to increase the

[19] Whether or not this unit feeds back to the hidden layer is largely inconsequential since its output remains constant throughout the encoding of a message, thus acting as an additional bias input.

[20] For example, for a fixed-length numeric code an encoder would send `#00$` for the concept 0 at 2-bit precision, and `#000$` for the same concept at 3-bit precision. With variable-length messages the encoder could send the message `#0$` for the concept 0, regardless of the precision.

number of strings that are properly terminated). Additionally, a small error ($2^{-2(k+1)}$ at k-bit precision) is added to the entire system for each string that is successfully communicated but improperly terminated, to differentiate between systems with different numbers of properly terminated sequences, yet similar error.

As before, 50 forwards and reversed systems were evolved to a maximum of 5-bit precision. Simulations were repeated under the same three different conditions used in study 2: forwards systems using a maximum of $2k$ symbols per message, and reversed systems using either $k+2$ or $2k$ symbols per message.

The performance of the systems in this series of simulations was significantly worse than in the fixed-length case (not surprisingly since it is necessarily a more difficult task). None of the reversed systems with a maximum message size of $k+2$ symbols attained 5-bit precision.[21] Results are summarised in Table 3.5. Again, the systems that reversed the encoders' output outperformed those in which the messages were transmitted unchanged.

Table 3.5: The level of precision obtained by systems in both the forwards and reversed conditions. The forwards systems, sending $2k$ symbols, attained an average precision of 2.90 bits. The reversed systems attained an average precision of 3.30 bits when transmitting $k+2$ symbols and 3.44 bits when transmitting $2k$ symbols.

Final Precision Reached	Forwards ($2k$)	Reversed ($k+2$)	Reversed ($2k$)
2	6	4	1
3	43	27	27
4	1	19	21
5	0	0	1
Total	50	50	50

As in the fixed-length case, both the forwards and reversed systems produced codes that could be placed in numeric order. The introduction of the EOS symbol created one novel difference: to be placed in numeric order the EOS often had to be assigned a value, indicating that it served as more than a syntax marker. A typical code found by a reversed $(k+2)$ system is shown in Table 3.5. This table also shows the code from the only forwards system that attained 4-bit precision, which is clearly of a different nature to the reversed system's code. While some reversed systems produced codes of a similar nature to this forward system's code, none were as regular. Codes from a reversed $(2k)$ system and the best forwards system are depicted graphically in Fig. 3.12.

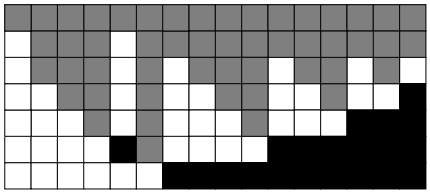
[21] Although one of two preliminary simulations did reach 5-bit precision.

Table 3.6: **Left:** Language for a variable length, reversed system at 4-bit precision, evolved with $k+2$ symbol channel length. Interpreting the EOS symbol (\$) as a value greater than 1, places the messages in numeric order. **Right:** Language for a variable length forwards system at 4-bit precision, evolved with channel length $2k$. These languages are also depicted graphically in Fig. 3.12(a) and (c).

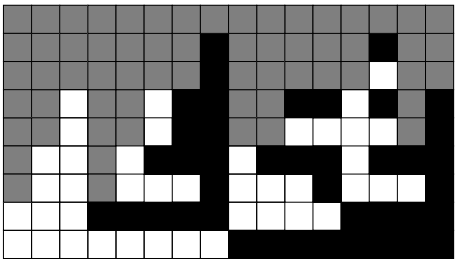
	Reversed $(k+2)$			Forwards $(2k)$	
Input	Message	Numeric	Output	Message	Output
0.0000	000000\$	0000002	-0.0037	00000000\$	-0.0019
0.0625	0000\$	00002	0.0631	0000000\$	0.0660
0.1250	000\$	0002	0.1251	0000001\$	0.1177
0.1875	00\$	002	0.2132	000000\$	0.1867
0.2500	010000\$	0100002	0.2384	000001\$	0.2506
0.3125	0\$	02	0.3309	00000\$	0.3126
0.3750	10000\$	100002	0.3829	00001\$	0.3790
0.4375	1000\$	10002	0.4279	0000\$	0.4382
0.5000	100\$	1002	0.4909	0001\$	0.5047
0.5625	10\$	102	0.5790	000\$	0.5629
0.6250	11000\$	110002	0.6366	001\$	0.6294
0.6875	1100\$	11002	0.6822	00\$	0.6868
0.7500	110\$	1102	0.7452	01\$	0.7532
0.8125	11100\$	111002	0.8219	0\$	0.8097
0.8750	1110\$	11102	0.8671	1\$	0.8646
0.9375	11110\$	111102	0.9371	\$	0.9380

Calculating the average OSE for the two sets of codes, as was done in study 2 (the fixed length case), shows a difference between the codes of the two conditions. Fig. 3.13 shows the average OSE for the reversed systems and for the single forwards system at 4-bit precision. Again, the two curves of the forwards system are closest together, indicating that information is distributed more evenly throughout the code (this result is not surprising considering the code, shown in Table 3.5). With only one forwards code it is difficult to draw conclusions, though there does appear to be a significant difference between the two sets of codes.[22]

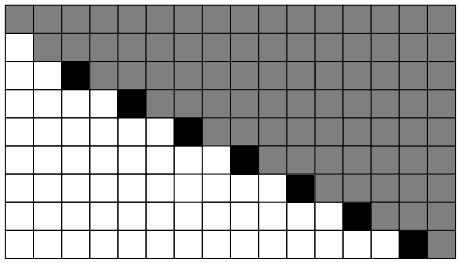
[22]The two sets of codes from systems at 3-bit precision were also compared using this test, but showed no significant difference.



(a) reversed, $k + 2$



(b) reversed, $2k$



(c) forwards

Figure 3.12: Graphical depiction of variable length codes for communicating values of 4-bit precision from (a) reversed systems sending $k + 2$ symbols (b) reversed systems sending $2k$ symbols and (c) forwards systems sending $2k$ symbols. In this figure, `1` is represented by black regions, `0` by white regions, with the grey areas showing where the encoder stopped producing output. Both codes were evolved using $2k$ symbol channel length, in contrast to the reversed code of Table 3.5 which was evolved with a maximum of $k + 2$ symbols in the communication channel. Note that languages (a) and (c) are described textually in Table 3.5.

3.6 Discussion and conclusions

The beginning of this chapter discussed the issues involved in creating a simulation framework for studying the emergence of structured communication systems, and proposed such a framework. This framework centered around RNNs as the communicative agents. This choice of agent raised many methodological issues. Foremost amongst these issues was the difficulty in searching through encoders in study 2. Encoders were evaluated by training a decoder with BPTT, a computationally expensive task. To reduce the number of these evaluations, encoders were first screened on their expressivity (step 4.b in Fig. 3.7). This cursory screening, while making the search task tractable, restricted the way in which encoder-space was searched. In particular, it precluded smaller but more easily learned languages.

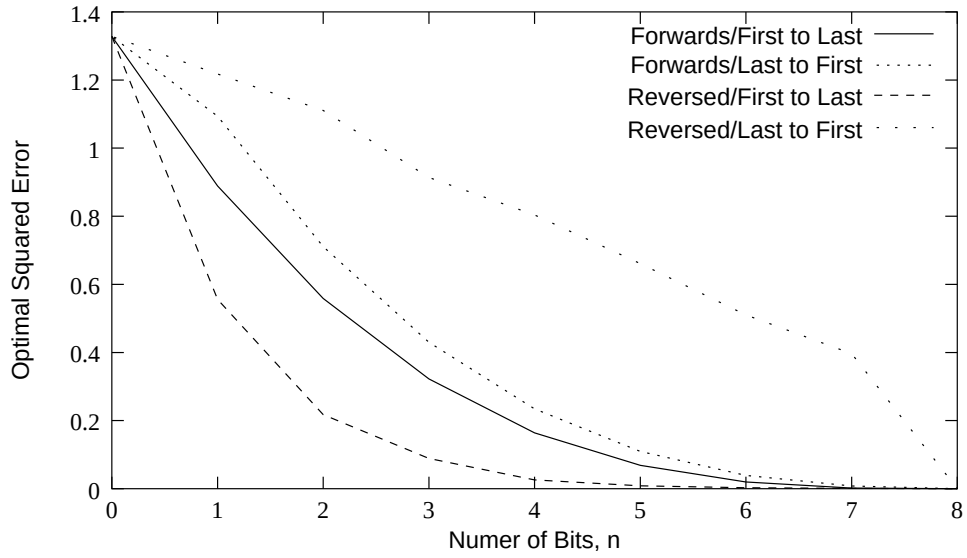


Figure 3.13: Optimal squared error obtainable from the one forwards system code for 4-bit precision, and the average obtainable from the reversed system codes at 4-bit precision. As in Fig. 3.11 the inner curves correspond to the forwards system and the lower curves result from calculating OSE first bit to last.

A further issue in searching through encoder-space was the problem of generating expressive encoders. Many heuristics (involving the choice of σ, the level of mutation) were used to try and produce more expressive mutant encoders. Even so, the search often took tens of thousands of iterations to find a more expressive mutant.

For the decoders too, there were methodological issues. The decision to take a staged learning approach interacted with the forwards/reversed simulation condition. In the reversed case, the hand-coded solution with the single hidden unit (Fig. 3.5b) is valid across different levels of precision, unlike the hand-coded solution with the single hidden unit in the forwards case (Fig. 3.5a) which has a weight dependent on the precision. The staged learning approach thus interferes with the forwards condition, but not the reversed condition, and is thus a potential confound. Simulations in the following chapters remove the staged learning aspect, suggesting that it was unnecessary.

The simulations of the first study (§3.3) established the differing biases of the encoders and decoders on the numeric encoding task. In the second and third studies (§3.4 and §3.5), languages were evolved so that these biases either opposed each other (the forwards case) or complemented each other (the reversed case). The structure of the forwards languages showed a compromise between the opposing biases

of encoder and decoder. The optimal squared-error (OSE) measurement showed that in the forwards languages, information was more evenly distributed throughout the sequence than in the reversed languages. This feature provides a compromise between placing the information at the beginning of the sequence (as preferred by the encoder) and placing it at end of the sequence (as preferred by the decoder).

In the framework presented, the adaptation of languages takes place in the absence of phylogenetic change — the biases of the encoder and decoder remain constant throughout a simulation. In reality, phylogenetic adaptation of language users may help to eliminate the differences between sender and receiver biases. However, the simulations demonstrate that adaptation of a language may be a contributing factor for the mediation of inconsistent biases. In Kirby's (2000) work there is no distinction between the constraints on the sender and those on the receiver, only a notion of what is easier to parse. In such a case, Kirby shows that the languages that proliferate are those that take on the forms that are easier to parse. Drawing a distinction between sender and receiver adds an extra source of bias to the system. When those biases are different, languages may take on forms that provide a compromise between the competing constraints.

Throughout this chapter the importance of learnability to the survival of a language has been emphasised — languages that match the innate learning biases of their users will tend to be more successful. In the simulations presented, the fitness of a language is determined by the success with which a decoder learns. Those languages that are "more learnable" by a decoder will have a greater chance of surviving. However, the encoder determines the language that is evaluated by the decoder. As demonstrated in §3.3, the encoder, in conjunction with the hill-climbing strategy, has a strong search bias for MSB-first languages. No doubt the introduction of a more realistic learning mechanism into the encoder would have important consequences that have not been considered. However, the simplified domain that we have presented contains the necessary elements to demonstrate the effect of conflicting biases on the shape of a language.

Chapter 4

Evolving Generalisable Languages

The previous chapter considered a language evolving in the presence of opposing biases from sender and receiver. However, those simulations did not take into account one of the defining aspects of language — generalisation. The language skills of agents were never tested on data to which they had not previously been exposed. The focus in this chapter is on the extent to which languages can adapt to be generalisable from a small amount of data.

4.1 The importance of generalisation

Generalisation is a core aspect of language, particularly with respect to syntax. While words and phonemes are restricted in number,[1] syntactic structures provide for an infinite range of expressions. Whereas the human learner can expect to be exposed to the entire range of words and phonemes, a substantial proportion of the utterances to which they will be exposed will be unique constructions. Without the ability to generalise from the finite set of utterances to which they have previously been exposed, the human learner would be unable to understand or produce novel utterances, thus restricting human languages to expressing a finite range of concepts. The most apparent form of generalisation used by humans is systematicity.[2]

The earlier work of both Batali (1998) and Kirby (2000) considered the issue of generalisation. In these studies the required level of generalisation is not comparable

[1] Ignoring morphosyntactic issues.

[2] Given a novel word, and its meaning, humans have few problems using it in a novel context. For example, if I tell you that 'to ploof' means to hit somebody over the head with a large, stuffed animal, you have no trouble understanding phrases such as, "Bob ploofed Jim," or, "Bob got ploofed."

with human learners. In Batali's simulations, learners are presented with 90 carefully selected meanings of the 100 possible. In Kirby's simulations, learners are presented with 100 meanings, randomly chosen from 100 possible meanings. While the chances of choosing 100 unique concepts are small, there is a high probability that a substantial number of the available examples will be chosen. In later studies (Batali, in press; Kirby, 1999b), where the number of available meanings is much higher, the degree of generalisation required is far more considerable. However, in both of these cases the learning mechanism severely restricts the kinds of generalisations that a learner may make because of an implicit assumption of *systematicity*. That is, the learning mechanisms have a (strong) search bias for systematic generalisation.

From the perspective of human languages, the assumption of systematicity seems a valid one since systematic generalisation is a ubiquitous phenomenon. However, while human languages may exhibit this form of generalisation, it may not be a necessary characteristic of languages *in general*. If, as stated earlier, the goal is to uncover the general principles behind language emergence, rather than the specific conditions that occurred during the evolution of human languages, then it seems inappropriate to restrict the learning mechanism to a specific class. It may be that there are viable languages where the required form of generalisation is unlike classical symbolic systematicity, or that systematicity is an emergent property of the dynamics of language transmission, independent of the type of learning mechanism involved. Thus, considering large-scale generalisation from a learner such as that used in the previous chapter (i.e., a recurrent neural network) will help to broaden the available evidence as to the necessary and sufficient conditions under which language may emerge.

As discussed earlier, generalisability of language is necessary for communication of a large (possibly unbounded) set of meanings. Thus, it seems likely that generalisability plays a role in glossogenetic selection (selection of linguistic features). Those linguistic structures that can be reliably generalised from fewer examples will be more easily acquired by learners, and thus more likely to be successfully transmitted between generations (Kirby, 2000). If languages do indeed adapt to aid their own survival, then they should be expected to adapt to more easily generalisable forms. What makes one form more easily generalisable than another? The generalisability of a language is primarily determined by the generalisation characteristics of the learning mechanism. As discussed in §3.1.2, the generalisation properties of a learner are determined by its inductive biases: the predisposition towards choosing

particular hypotheses from those that are consistent with the data. We might expect then, that languages will evolve to exploit the inductive biases of their learners to enhance their generalisability.

In the simulations of both Kirby (2000) and Batali (in press), the evolved languages facilitated systematic generalisation, as would be expected given the learning mechanisms employed. However, these learning mechanisms were tightly constrained as to the type of grammar that would be generalised from a given set of data. That is, learners made very consistent generalisations. The same can not be said of recurrent neural networks (or neural networks in general), where the outcome of learning may be sensitive to the initial weights of the network (Kolen and Pollack, 1990) or the selection of training data. Small differences between networks may lead to radically different outcomes for learning. An open question is whether a language can evolve to assist generalisation in such a fickle learner. The first study in this chapter demonstrates that languages can indeed facilitate an impressive degree of generalisation in such a learner.

In the traditional view of learning, where the task is fixed, the difficulty of the task itself is a significant determinant of the likely accuracy of generalisation. In the simulations considered thus far, this problem has been largely avoided since the task (in the form of the language) is free to vary. There remains however, a higher-level task, that of end-to-end communication. That is, the combined sender-receiver system has the overall task of accurately communicating a concept. The basic case is where the desired output of the receiver mimics the input to the sender. However, other schemes are possible.

It is possible to envisage a situation where the communicative goal of the sender is to convey some *aspect* of a given concept, rather than the concept itself; where the desired output of the receiver is some function of the input to the sender. In the second study in this chapter we require that the output of the decoder network approximate some function of the encoder's input (that is, where $y = f(x)$ in Fig. 3.2). By considering a range of such functions, it is possible to analyse whether a language can evolve to support specific generalisation requirements, that is whether the language supports a particular *form* of generalisation. The extent to which the generalisation performance of the learners is due to the assistance given by the languages, rather than the sophistication of the learners themselves can be tested by 'migrating' the languages. That is, by moving a language evolved in a system with one communicative goal to a system with a different goal and measuring the success

with the 'migrant' language facilitates (learned) communication on the 'native' task.

While the learning mechanism is the source of the generalisation properties, a variety of other factors can contribute. The most apparent of these is the data itself. Obviously, as the amount of (labelled) data given to a learner increases, the data should begin to overwhelm the inductive biases of the learner since less generalisation is required. The manner in which training data is selected may also play a role. Intuitively, one would expect that to be more reliably generalisable, training data should be sampled from a broad distribution across the space. However, the most desirable distribution of data may depend on the properties of the learner and task (Elman, 1993). By analogy with the argument for language adaptation to the learning mechanism, this notion suggests that languages may adapt to be learnable from a specific set (or distribution) of examples. That is, languages may evolve to become easily generalisable from just those examples to which a child is likely to be exposed in its environment. The third study in this chapter considers a fixed learning environment, where each individual learns from the utterances produced from an unvarying set of concepts, thus giving language the opportunity to adapt to a stable learning environment.

To summarise, the aim of this chapter is to explore the issue of generalisation as it pertains to the evolution of language-like communication systems. Particularly, we consider whether a language can adapt to become learnable (and generalisable) by a general-purpose learner from sparse data. This work extends that of Batali and Kirby by considering an alternative learning mechanism (and task) that has markedly different learning properties, namely that generalisation is less predictable. The simulation results add weight to the claim that an appropriately constructed language can enhance the generalisation abilities of learners by exploiting their inductive biases. Studies two and three examine the role that the learning environment may provide in facilitating generalisation. The second study in this chapter considers variations of the communicative task. A variety of languages are evolved for different communicative tasks with the simulation results indicating that languages may be capable of adapting to facilitate specific forms of generalisation. The third study considers the conjecture that language adaptation may exploit the fact that learners' environments have much in common. The results are inconclusive, but raise some interesting possibilities regarding languages and the learning environment of their users.

4.2 Simulation design

4.2.1 Communication task

The simulation approach used in this chapter is broadly similar to that used in the previous chapter. Two recurrent neural networks, one acting as sender, the other as receiver, attempt to communicate a "concept" represented by a point in the unit interval, $[0, 1]$ over a symbolic channel (see Fig. 3.2). The major change in the communicative task introduced in this chapter is the choice of alphabet. In the previous chapter, messages were (fixed or variable-length) sequences of symbols from an alphabet of size two ($\Sigma = \{0, 1\}$), bounded by start-of-sequence and end-of-sequence markers ($ and #). These symbols were represented by a one-hot vector encoding (i.e., $[1, 0, 0, 0]$, $[0, 1, 0, 0]$, $[0, 0, 1, 0]$, $[0, 0, 0, 1]$). Such a restricted alphabet makes it difficult to convey much information with short messages, and imposes constraints on the types of languages that are feasible. Consequently, the alphabet in this chapter has been expanded to ten symbols, as well as an additional end-of-sequence marker. The start-of-sequence token has been omitted as it plays no significant role — resetting the weights of the decoder network between utterances is sufficient.

The one-hot encoding scheme used in the previous chapter would require each symbol to be an eleven dimensional vector, one dimension for each of the ten symbols and end-of-sequence. From a learning perspective it is not possible to generalise this form of representation (i.e., knowing how to use one symbol gives no information on how to use the other symbols) so an individual must be exposed to each symbol during learning to be able to correctly interpret it. While this behaviour accords closely with the notion of a symbol, it is impractical for the simulations in this chapter which use very small training sets, making it probable that learners will not be exposed to the entire range of symbols. Thus, the symbols in this chapter are more 'informationally dense' than those used in the previous chapter. An alternative approach is consequently taken in converting the continuously valued network outputs into symbols.

The encoder network has five (continuously-valued) output units that must be transformed into one of eleven symbols, denoted A, B, ..., J and $ (where $ is again the end-of-sequence marker). These symbols correspond to the binary vectors $[1, 1, 0, 0, 0]$, $[1, 0, 1, 0, 0]$, ..., $[0, 0, 0, 1, 1]$ and $[0, 0, 0, 0, 0]$ respectively: each of the message symbols are five-bit patterns, with two bits on and three off, with the end-

of-sequence marker being all zeros. Network outputs are transformed into these patterns in the following way. If none of the encoder network's output units has an activation greater than 0.5, then `$` is sent. Otherwise, the two highest activations are saturated to a value of 1, the remainder set to 0, and the corresponding pattern sent ('two-winners take all'). Hence, to encode a concept $x \in [0, 1]$, the encoder network is presented with a sequence of inputs $(x, 0, 0, ..)$. At each step, the output units of the network assume one of eleven states: all zero (`$`) or exactly two units on (`A`, ..., `J`). This transformed output activation vector is then sent to the decoder. If the `$` symbol is produced, propagation of activation through the encoder network is halted. Otherwise propagation continues for up to five steps, after which the output units assume the zero (`$`) state, such that the end-of-sequence marker is always sent.

The general architecture for the networks remains the same from the previous chapter, with five hidden units used for both encoder and decoder. To conform to the change in symbol encoding, the encoder network has five output units and the decoder network five inputs.

4.2.2 Interaction dynamics

The results from the previous chapter indicated that, due to conflicting constraints of the encoder and decoder, it is easier for the decoder to process strings that are in the reverse order to that produced by the encoder. In this chapter, the primary aim is to test the extent to which a language can exploit the inductive biases of learners, so it seems reasonable to simplify the learning task as much as possible. Consequently, the input to the decoder is taken to be the reverse of the output from the encoder, (except for `$`, which remains the last symbol). For clarity, utterances will be written in the order produced by the encoder. Each input pattern presented to the decoder matches the output of the encoder — either two units are active, or none are. As before, the decoder network is trained with BPTT to produce the desired value on presentation of the final symbol in the sequence (which will always be `$`).

The hill-climbing strategy used in the previous chapter is again applied to the encoder network (see Fig. 3.7), with some changes to the way in which learning is used to evaluate the network. Whereas before decoders were trained using a staged learning approach, with all values of a given precision appearing in the training set, this chapter's focus on generalisation requires an alternative approach. Since the encoder's input space is continuous and impossible to examine in its entirety, the

input range is approximated with 100 uniformly distributed examples from 0.00 to 0.99. From this space of 100 concepts, decoders are trained on a small, randomly selected subset for 400 epochs. After training, decoders' language abilities are determined by calculating the sum-squared error across the entire range of concepts. The evaluation of encoder networks in the hill-climbing algorithm (Fig. 3.7) is thus: (a) whether the mutant encoder produces a greater number of unique messages; and (b) whether a decoder with random initial weights has, after training on the output of the mutant encoder, a lower sum-squared error than the decoder trained on the output of the current champion.

The simulation design addresses the issue of generalisation by utilising a training set that is smaller than the entire space. The hill-climbing algorithm, defined on the encoder, searches for a network that produces a language that a random decoder can generalise from a small training set. The existence of such an easily generalisable language would demonstrate that languages that evolve to exploit the biases present in their learners may not need an additional, specifically tailored, innate language competence. The search process is primarily aimed at finding a language that is easy to learn, rather than at finding the encoder network itself. The results from the previous chapter suggest that languages constructed by reversing the output of the encoder provide likely candidates. That is, the search space that is formed by the languages that are the reversed output of encoder networks is expected to be suitably biased towards producing easily generalisable languages. This space is also amenable to search via the weights of the encoder network.

4.3 Study 1: Evolving for generalisability

It seems unreasonable to expect that, from just five training examples, a decoder could successfully generalise to an entire language. Consider again the regression problem, shown in Fig. 3.1. It is obvious that as the number of labelled examples decreases, the problem becomes harder. For a learner with low model bias, the number of consistent hypotheses may be immense. Such a learner, when coupled with an algorithm with little search bias, could be expected to find any of the many alternative hypotheses. It is thus expected that the learning task should be hard for the recurrent neural network learner used here. (Note that for the language task, simple linear interpolation is not possible since generalisation is based on the symbolic structure of the language, rather than the properties of the concept space.)

Ten encoder networks were evolved with the hill-climbing algorithm for 10000 generations.[3] All ten runs used encoders and decoders with five hidden units. At the conclusion of the evolution phase the final languages were extracted and used to train 100 new random decoders under the same conditions as during evolution (i.e., five examples, 400 epochs). These additional 100 learners provide a more accurate measure of the learnability of the evolved language than the single learner used in the evolution phase. Hence, it is these learners that are used to evaluate the ease with which the evolved languages can be learned.

4.3.1 Study 1: Results

To provide a summary of the results, it is convenient to define a performance criterion that can be used to categorise whether or not a decoder has adequately acquired a language. The choice of criterion is arbitrary, but inspection of a variety of trained decoders suggested that a sum-squared-error across the 100 points in the space of less than 1.0 represented a creditable solution to the communication task, given its difficulty. Such a decoder is defined to have *learned* the language. Furthermore, a language is defined to be *reliably learnable* when at least 50% of random decoders are able to learn it (to the above criterion) within 400 epochs.

Under the measure outlined above, all of the evolved languages were learnable by some decoders with the 'hardest' language having only 20 successful learners, and the 'easiest' having 72 successful learners (the mean being 48). Of the ten languages evolved, five were reliably learnable. Encoders employed, on average, 36 unique utterances (minimum 21, maximum 60) to communicate the 100 points (i.e., all of the evolved languages had a substantial degree of homonymity). There was a strong correlation between the size of the language and the number of successful learners ($r = 0.83$). That is, the larger languages were more learnable.

The language abilities of the evolved encoders and trained decoders are best demonstrated pictorially. However, it is impractical to provide a complete description of each of the ten trials. Thus, the remainder of this section provides a detailed examination of one of the ten trials. The language that emerged from this trial was one of the more learnable (the third best of the ten), but the general behaviour is

[3] As before, one generation represents the creation of a more variable, mutated encoder and the subsequent training of a decoder. More generations were possible in these simulation than those in the previous chapter due to the decreased amount of learning, both in terms of the amount of learning data and the number of training epochs.

qualitatively representative of all ten trials.

The structures of the evolved languages makes it possible to depict them as trees. This representation is similar in spirit to that used in the previous chapter (e.g., Fig. 3.4) but is more suitable for languages with larger alphabets. Furthermore, the set of training examples form a sub-tree of the complete language tree. Showing both the training set sub-tree (see Fig. 4.1) and the complete language tree (see Fig. 4.2) highlights the inherent difficulty of the generalisation task: decoders must complete the entire tree structure from a fragment of the same size as that highlighted.

Since the communicative goal is the accurate transmission and reception of a point, plotting the decoder's output against the encoder's input should yield a curve similar to $y = x$. Furthermore, it is possible to see how the dynamics of a decoder construct this approximation from successive symbols (Fig. 4.3). This series of figures highlights the recursive nature of the language (and the way it is processed). The decoder's output does not monotonically approach the desired output. Rather, similar sub-structures are constructed across the space and are differentiated by symbols that are received later. This effect is most apparent in Fig. 4.3(d) where similar structures appear in both halves of the space. In other networks, the effect is more marked.

This section has to this point considered only the final language produced by the hill-climbing algorithm. Also of interest is the progress that the system makes from an initial language, produced by a random encoder, to the final language. To this end, the most relevant statistic is how the hill-climbing algorithm improved the language over time (see Fig. 4.4). This figure shows a monotonic improvement over time, as guaranteed by the hill-climbing algorithm. However, the languages produced by the encoders have been subjected to a very unreliable evaluation function: learnability by a single, random decoder from five random training examples. The 'true learnability' of the language is thus unlikely to be precisely the same as the estimate of the learnability made during the evolution phase.

The large number of languages that are evaluated will also cause the learnability of a language to be over-estimated. For a given language, the performance of all possible learners with all possible training examples will have some distribution of errors. The most desirable languages are those in which all learners do well (i.e, a distribution with small variance and with a mean error close to zero). In effect, the hill-climbing algorithm used in these simulations estimates the distribution from a single example — the decoder trained during the evolution phase. Since this

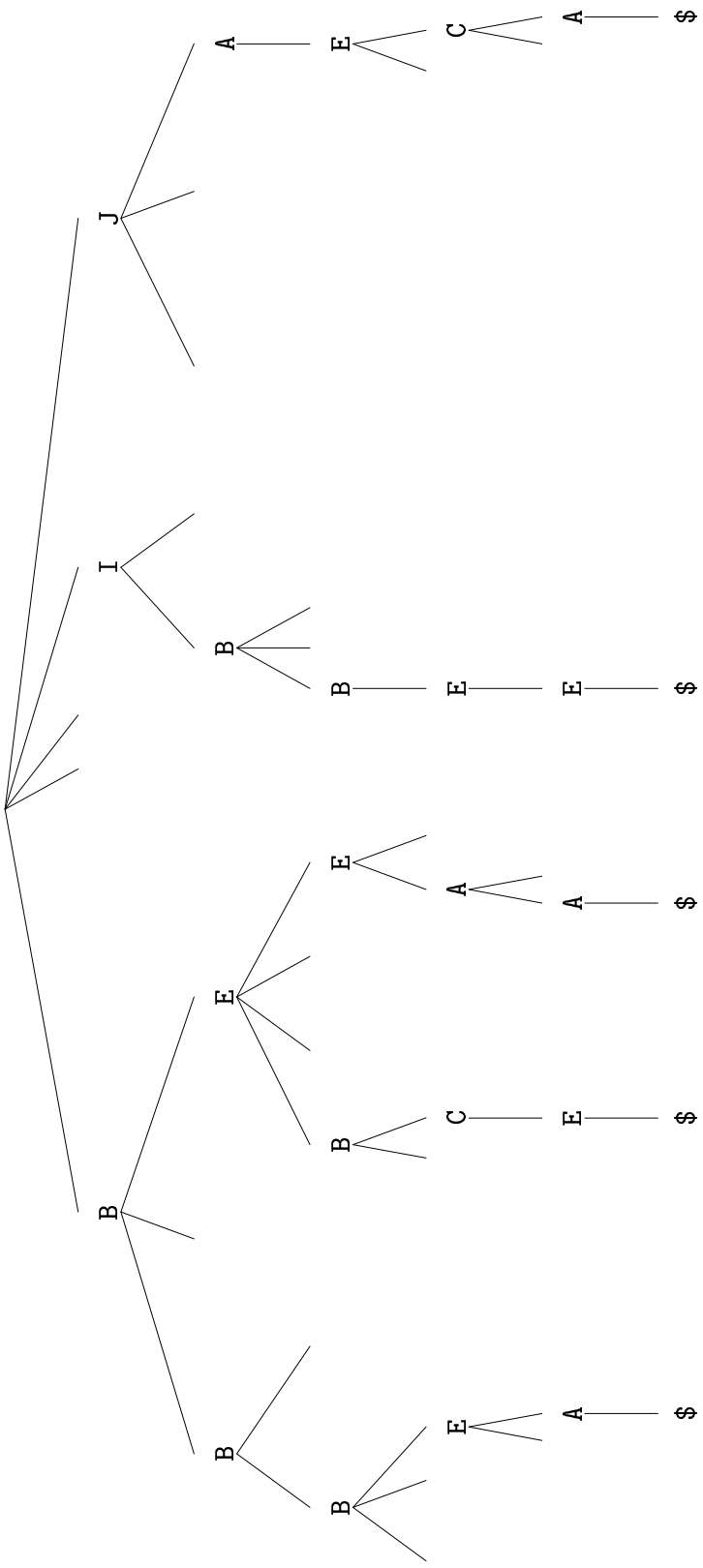


Figure 4.1: Language subtree presented as training data to learner. This tree represents the training data, as produced by the final champion encoder that is given to a learner. (For description of tree structure, see caption of Fig. 4.2.) The five examples provide only a sparse coverage of the input space, with the learner required to generalise to the strings that form the remainder of the tree, shown in Fig. 4.2. This generalisation requires the learner to correctly interpret symbols in novel positions. For example, C appears in only the fourth level in the training set, but appears in the second, third, fourth and fifth levels in the complete language.

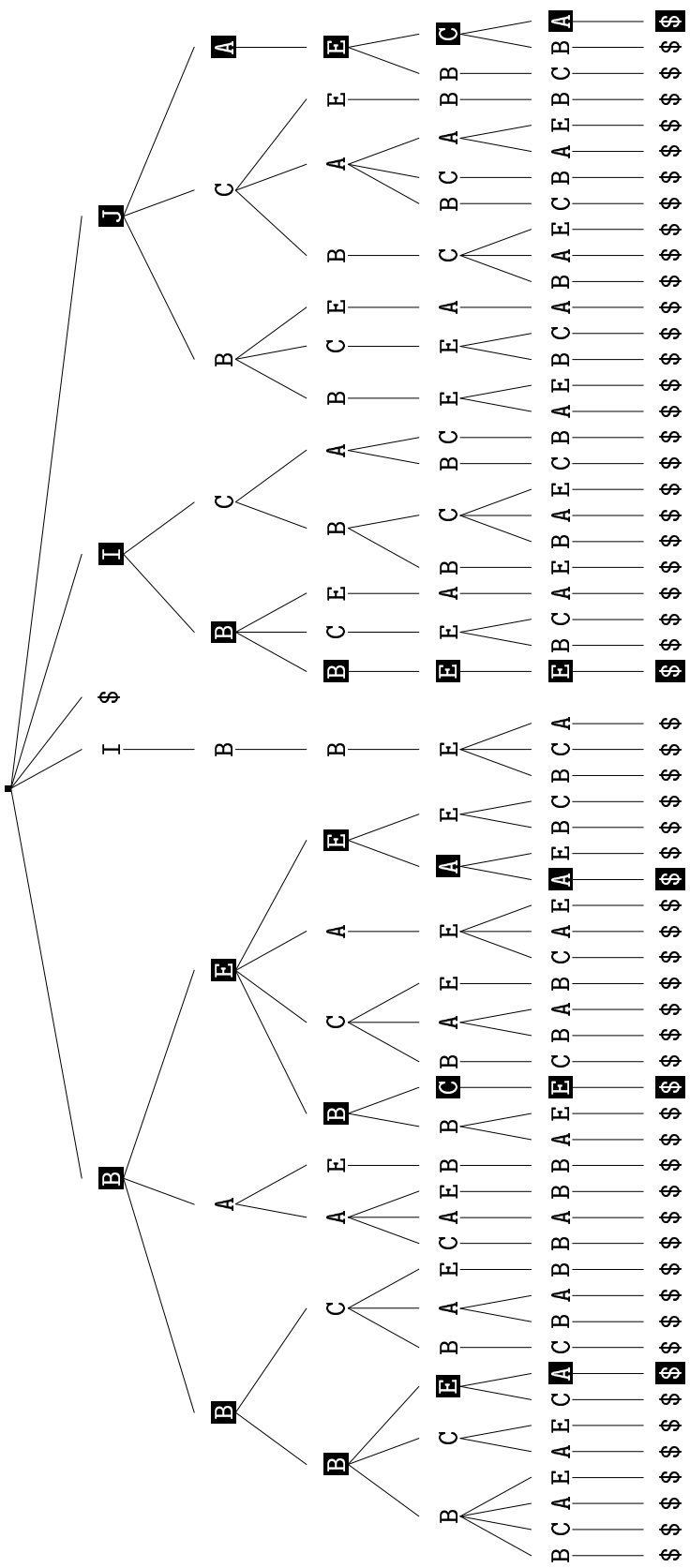


Figure 4.2: Hierarchical decomposition of the language produced by an encoder, with the first symbols produced appearing near the root of the tree. (Note that the decoder is presented with strings that are the reversed output of the encoder.) The ordering of leaves in the tree represents the input space, smaller inputs being encoded by those utterances on the left. The examples used to train the decoder are highlighted and are also shown in Fig. 4.1. The decoder must generalise to all other branches. To learn the task, the decoder must generalise systematically to novel states in the tree, including generalising to symbols in different positions in the sequence. Some decoders have even successfully generalised to novel symbols after learning from pathological training sets. Such generalisation is possible because of the style of symbol coding.

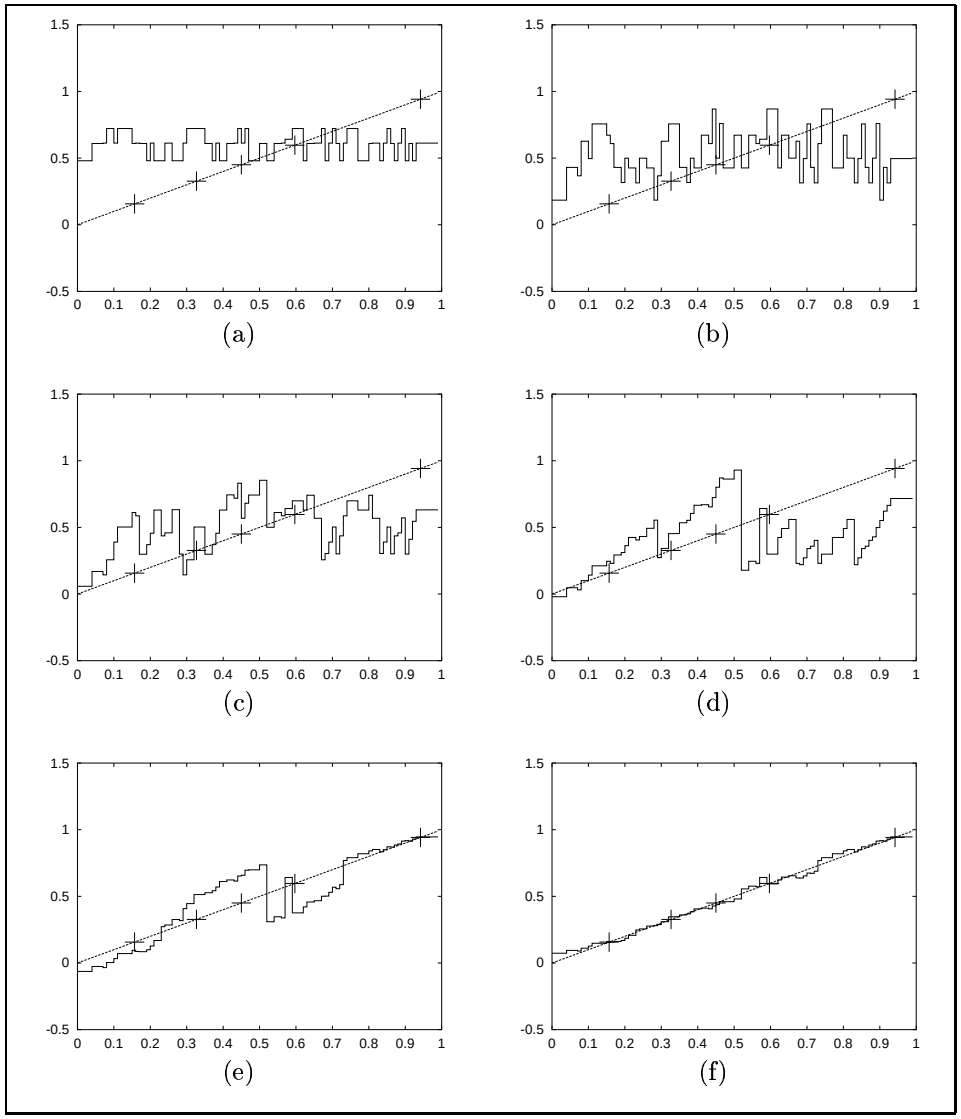


Figure 4.3: Decoder output after seeing the first n symbols in the message, for $n = 1$ (a) to $n = 6$ (f) (from the language in Fig. 4.2 and the training examples from Fig. 4.1). The x-axis is the encoder's input, the y-axis is the decoder's output at that point in the sequence. The five points that the decoder was trained on are shown as crosses in each graph. After the first symbol (`A`, `B`, `C`, `E` or `$`), the decoder outputs one of five values (a) with more states after successive symbols. Subsequent symbols in each string specify finer gradations in the output. Note that the output is not constructed monotonically, with each symbol providing a closer approximation to the target function, but rather recursively, only approximating the linear target at the final position in each sequence. Structure inherent in the sequences allows the system to generalise to parts of the space it has never seen. Generalisation is not based on interpolation between symbol values, but rather on their compositional structure.

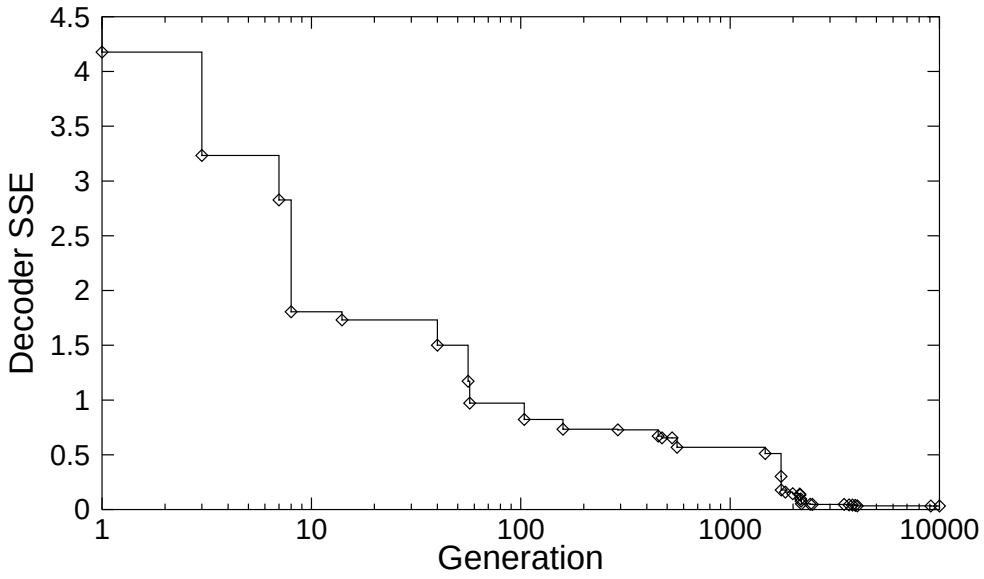


Figure 4.4: Language improvement during the course of evolution. Each 'step' in this figure corresponds with the ascendancy of a champion encoder/decoder pair (note the logarithmic scale on the x-axis). The fitness of this pair, (shown on the y-axis) is the squared error after training, summed across all 100 test concepts. (Recall that when this value is less than one, the decoder is said to have adequately learned the language according to the definition given earlier.)

algorithm searches for the single best learner, the learner that it finds is likely to be from the best part of the distribution of all learners on the language. Although the estimate of learnability provided by the single learner is unreliable, it is expected that the better learners will tend to come from better distributions. That is, those languages whose single trained decoder has very low error should also tend to have reasonable distributions of errors across all learners.

Given the unreliability of the learnability measure in Fig. 4.4, it seems pertinent to test the learnability of the networks with a more reliable measure. Such testing can be done by taking each of the 34 'champion' encoders and training 100 new decoders on the language produced (see Fig. 4.5). The resulting estimate of the 'true learnability' curve is not monotonic decreasing, reflecting the unreliability of the learnability measure used in Fig. 4.4. However, the general trend of the curve is towards more learnable languages, as expected.

Together, Figs. 4.4 and 4.5 show the learnability statistics of the language as it evolves, but fail to be informative about the changing structure of the language. In Fig. 4.5, five positions are highlighted on the graph (labelled $\boxed{1}$ through $\boxed{5}$).

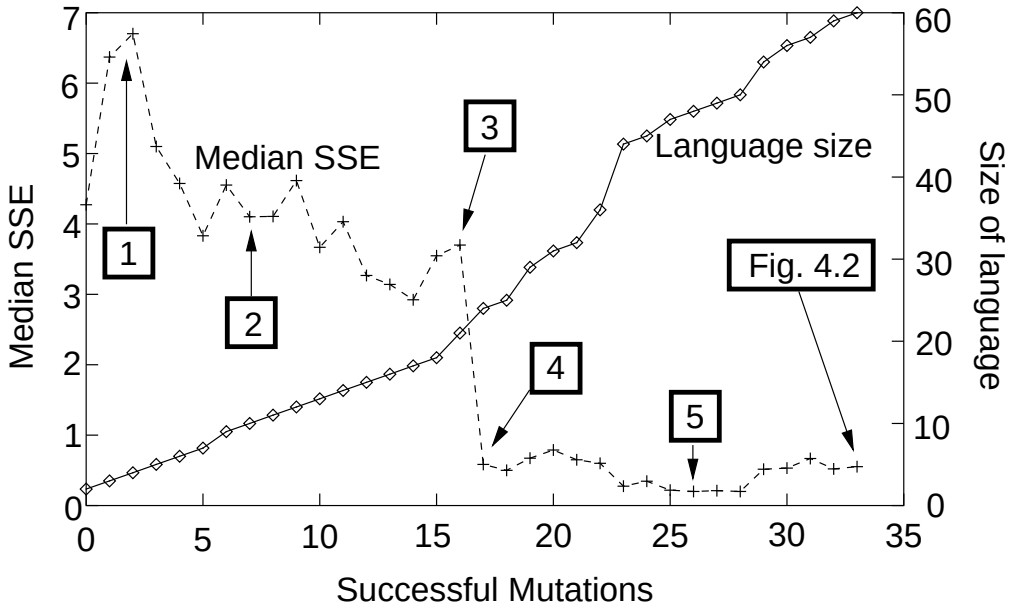


Figure 4.5: Estimate of true learnability of languages during the course of evolution. Each point in this figure corresponds to a champion encoder in Fig. 4.4 (which appears as a 'step') with the x-axis collapsed across the generations. The language produced by the encoder is tested by training 100 random decoders on five random examples (chosen differently for each decoder) and then testing generalisation performance across the range of 100 concepts. The y-axis thus plots the median squared error summed across the 100 concepts. The median is used rather than the mean since the distributions have a substantial positive skew. Recall that languages with a median squared error of less than one are deemed reliably learnable. Also shown is the number of unique utterances produced across the range of concepts. The languages at the five highlighted points are shown in Fig. 4.6 (see text for details).

The languages produced by the champion encoder at each of these five positions are shown in Fig. 4.6. The final language from this trial appeared in Fig. 4.2.

1 At the beginning of the trial there are very few unique strings, resulting in a highly homonymous language. Learnability is poor as a consequence of the lack of information in the language.

2 After the seventh successful mutation, the language has grown but remains quite irregular. A completely different set of symbols are being used to those that were in use earlier, and the language shows some similarities with the final language (Fig. 4.2). The same string (`I$`) is used to encode two non-contiguous regions, separated by the empty string (`$`).

[3] Successful mutation 16: the language continues to grow, but remains inadequately learnable. The same general structure is retained, although the `F` symbol makes a reappearance (it appeared earlier at [1]).

[4] Successful mutation 17: in a single mutation the language undergoes massive regularisation. Learnability sees a substantial increase. Shorter strings disappear, and an entire branch of the tree (headed by `J`) disappears. Some irregularities remain, particularly the symbols `F` and `A` which occur only rarely.

[5] Successful mutation 26: the learnability of the language improves even further as does its regularity. Apart from the lone `I` node (which will appear in half the language given its position), the tree is comprised of `B`, `C` and `E` nodes. The order of symbols is readily apparent, with `B` always the left-most child node (when it appears), followed by `C` then `E`. This ordering is consistent at each level of the tree.

[Fig. 4.2] Successful mutation 33: the learnability of the language, as determined by the more reliable measure, worsens. Irregularities return. Much of the structure from [3] has returned, most notably the `J` branch. The `A` symbol appears throughout much of the tree (as it does in [4]), where in [5] there appeared only `B`, `C` and `E`.

4.3.2 Study 1: Discussion

Given the paucity of training data, it is surprising that a language with any degree of learnability could be found. While the reliability with which decoders learned was not spectacular, they certainly were far more successful than anticipated given the difficulty of the task. The five training examples provide only a sparse coverage of the space, and their random sampling leaves a significant possibility of a pathological distribution (e.g., if all five example concepts are taken from a very small region of the space).

Additional tests were performed on the final languages where learners were provided with a greater number of training examples, either ten or twenty. As expected, with more training examples (and hence diminished likelihood of unrepresentative sampling), learners fared far better. When learners were given ten training examples, the ten languages were on average learned by 86% of decoders. With twenty training examples, the ten languages were on average learned by 97% of decoders.

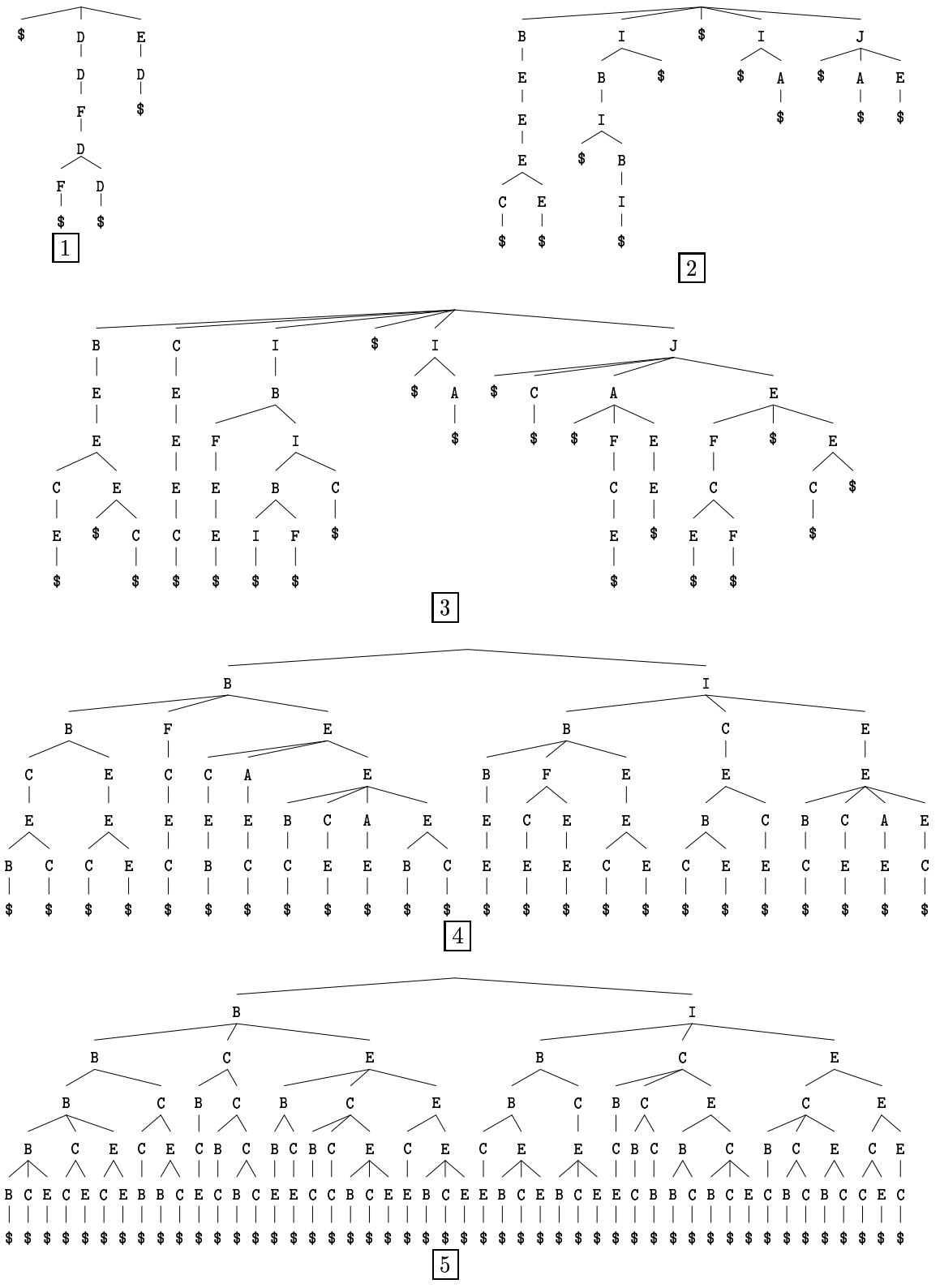


Figure 4.6: Languages at five points during course of evolution as marked in Fig. 4.5.

As the amount of training data provided to learners increases, their performance also increases. An interesting question that arises for consideration in future work is thus the trade-off between the amount of training data and the degree of success (i.e., the point at which is it preferable to forego increases in learnability for reduced training).

Further simulations attempted to establish whether evolved languages were suited to being learnable from a specific number of examples. That is, whether the language was more easily learnable from M examples than N examples ($M \neq N$). This relationship was tested by taking languages that had been evolved to be learned from M examples, L_M, and languages that had been evolved to be learned from N examples, L_N, and comparing the learnability of L_M and L_N from N examples. No relationship could be established between the size of the training corpora during evolution and the size of corpora during testing.

The sparse, random sampling of the training data caused by the small training corpora was not the only potential source of difficulty. Each of the 100 decoders used to test the final language had different initial weights, creating an additional source of variability and uncertainty in the learning process. Why then, were learners so successful when learnability theory might suggest otherwise? Inspection of the decoder networks' performances on the training examples (by examining the errors at the training examples in graphs such as that shown in Fig. 4.3) revealed, not unexpectedly, that in most cases errors at the training examples were small. BPTT had no significant problems minimising training error (i.e., finding a consistent hypothesis), but this observation does not explain why generalisation was so successful.

Restricting the search for learnable languages to those produced by an encoder network provides part of the answer: sampling languages from this subspace is far more likely to produce a learnable language than sampling from the entire space of possible languages. However, two questions remain. Why are even these languages learnable (i.e., why should *any* language be learnable if RNNs are such fickle learners)? Why should this subspace in particular be learnable, rather than any other?

The answer to the first question appears to be that, at least for the task under consideration, BPTT has a considerable search bias. The previous chapter demonstrated the ease with which a decoder could be constructed (Fig. 3.5). The dynamics of the hidden unit in this network are straightforward, being dominated

by a single attractor.[4] The dynamical properties of a network determine its general behaviour. If the dynamics that are established in a network by minimising error on training data correspond to the dynamics required for the whole problem, then generalisations are more likely to be accurate. Thus, a plausible explanation for the consistently correct generalisation in the decoder is that the appropriate dynamics are easy to establish in the decoder network. BPTT starts with a network with small random weights. These networks are likely to already have an attractor dynamic (Wiles and Elman, 1995). The five training examples are then sufficient to set the required parameters for driving the dynamics, thus leading to good generalisation performance.

An intuitive answer to the second question — that of why encoders produce learnable languages — follows from a similar consideration of the dynamics. Since both the encoder and decoder network are so closely related (both are recurrent neural networks), they are governed by similar computational mechanisms. The types of dynamics that can be established in the encoder can also be established in the decoder. The fact that the two processes are related provides some benefit. Consider again the non-linear regression problem of Fig. 3.1. There will be some advantage to choosing a learner from the same class as the target function: if the labelled data is known to be generated by a polynomial, it makes sense to try to fit that data with a polynomial curve. The dynamics of the encoder constructed in the previous chapter (Fig. 3.3) are dominated by a repelling fixed point, and two associated attracting fixed points. This dynamic is effectively the inverse of the dynamic found in the decoder. The correspondence between encoder and decoder implies that for any easily-found encoder, there should be an associated, easily-found decoder.

While recurrent neural networks may have relatively small *model* bias, the results of this study suggest that, when combined with BPTT, they may have a significant *search* bias. On the decoding task under consideration, the search space provided by the encoder network assists in exploiting this search bias so that learnability is boosted beyond what might be expected. If a real learned communication system were to have similarly exploitable biases then we might expect a similar outcome:

[4] That is, in the absence of any input, the activation of the hidden unit will tend toward some fixed value over time, regardless of the starting activation value. More substantial coverage of the dynamics of RNNs can be found elsewhere in the literature (Wiles and Elman, 1995; Tino et al., 1995; Rodriguez et al., 1999; Bodén et al., 1999, for example,), and is outside the scope of this thesis.

that a learner's guesses about the language would tend to be correct since the language could evolve to match the intuitions of its learners.

4.4 Study 2: Evolving for different generalisations

The first study demonstrates that for the recurrent neural network learner there exist languages that can be generalised from few examples. Study 2 probes the role that the language plays in generalisation. Particularly, the simulations address the question of whether a language can support different generalisation requirements. Study 1 §4.3 considered the case where the decoder's required output was the same as the encoder's input, yielding the approximation to the line $y = x$ in Fig. 4.3(f). Given a set of pairs of concepts ($\mathbf{x}$) and utterances ($\mathbf{U}$), $\{(x_1, U_1), \ldots, (x_n, U_n)\}$ (i.e., the labelled data of the training set), the decoder was required to generalise the relationship for all utterances, $\mathbf{U} \rightarrow \mathbf{x}$. In the simulations presented in this section, decoders are instead given a set $\{(f(x_1), U_1), \ldots, (f(x_n), U_n)\}$ and are required to generalise the relationship, $\mathbf{U} \rightarrow f(\mathbf{x})$.

The introduction of the function $f(\cdot)$ changes the nature of the relationship that the decoder is required to generalise. As an extreme example, consider the case where $f(x) = k$ for some constant, k. With such a function, the language should be trivially generalisable. However, if the language has many different utterances, then the learner will be required to generalise each different utterance to the same (output) concept. If the language has only a single utterance, then learning is trivial. The opposite extreme is where $f(x)$ is defined such that the outputs for similar values of x are unrelated (e.g., where $f(x)$ produces a randomly chosen value). In this case, languages such as those found in the previous section cannot be successfully generalised by the decoder. What is required in this case is a re-ordering of the utterances so that utterances are ordered with respect to $f(x)$ rather than x (i.e., where similar utterances encode similar values of $f(x)$ rather than x). In these two examples, the structure of the language can significantly alter the difficulty of the generalisation task. The choice of f for a learner will be referred to as the *world*.

The simulations thus aim to demonstrate that the generalisation performance found in §4.3 does not stem from an omnipotent learner, rather, that the structure of the language itself facilitates learning. For this assertion to be true, languages that have been chosen so as to be learnable in one world should not be as easily learned in an alternative world, unless of course the two worlds are in some way

homologous (e.g., $f(x) = x$ and $f(x) = -x$).

Two sets of ten languages were evolved, each set using a different world: either the same world as in the first set of simulations, or a set of random steps (see Fig. 4.7). On the completion of the evolution phase, the final languages were tested for learnability in four different worlds (including the one in which they were evolved). The four worlds — identity, random-step, sine and cubic — were chosen so as to vary in monotonicity and continuity.

- The identity world is both monotonic and continuous.
- The random-step world is non-monotonic and non-continuous.
- The sine world is non-monotonic and continuous.
- The cubic world is monotonic and continuous like the identity world, but differs by having a non-constant derivative.

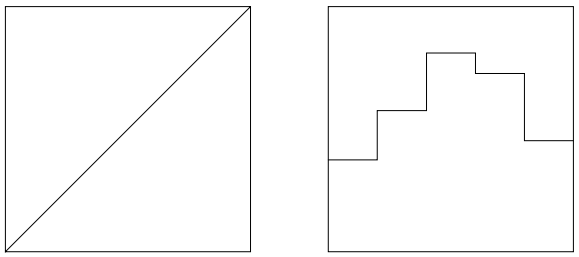
Again, testing was performed by training 100 new random decoders. Languages were evolved in the same manner as in §4.3, with the exceptions that ten training examples were used rather than five, and the hill-climbing algorithm was run for only 1000 generations.

4.4.1 Study 2: Results and analysis

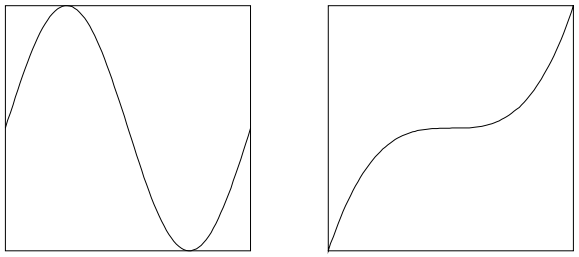
As expected, languages were substantially more learnable in their 'native' worlds (see Table 4.1). Languages evolved for the identity mapping were on average learned by 64% of decoders trained on the identity task compared with 0%–29% in the other worlds. Languages evolved for the random-step task were learned by 60% of decoders trained on the random-step task but only 0%–24% when trained in other worlds. Decoders generally performed poorly on the cubic function, despite some similarities with the identity function, and no decoder learned the sine task from either set of evolved languages.

Some additional tests were performed with alternative functions for both evolution and testing, including two quadratic functions (one monotonic, the other 'U' shaped). In every case, learners in the native world of the language outperformed learners from non-native worlds.

Presumably, the learners in the native world of the language outperform learners in non-native worlds because the structure of the language has adapted to be generalisable in a specific way. That is, the structure of the language captures properties



(a) identity (b) random-step

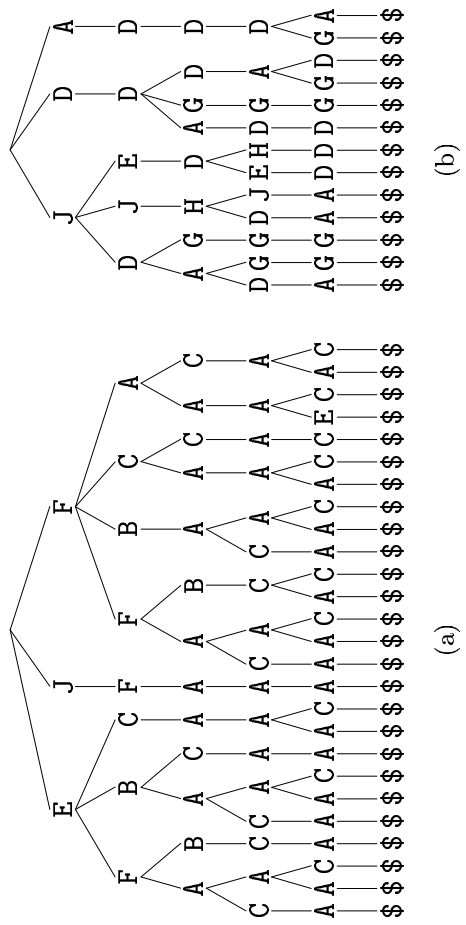


(c) sine (d) cubic

Figure 4.7: Four alternative 'worlds' that impose different generalisation requirements on learners. These functions show the desired relationship between encoder input, shown on the x-axis, and decoder output, shown on the y-axis. The identity world (a) is the same as that used in §4.3. Languages were evolved for learnability in either the identity or random-step worlds and were tested for learnability in all worlds (i.e., (a) and (b) are used for both evolution and testing, and (c) and (d) are reserved for testing).

of the structure of the environment (world), which learners then exploit. Surprisingly, an initial inspection of the two sets of languages (identity and random-step) revealed no obvious structural features that would indicate one was more suited to a random-step world than the other (see Fig. 4.8). Certainly, the figure shows a clear difference between the relative sizes of the languages, with the identity language having twice as many distinct strings as the random-step language. This result is to be expected — unlike decoders in the identity world, random-step decoders only have five different output values. Consequently, languages with many different utterances are probably suboptimal (but inevitable given the requirement for constantly larger languages in the hill-climbing algorithm, Fig. 3.7).

Differences beyond the sizes of the languages are revealed by consideration of the structure of the language with respect to the structure of the environment (Fig. 4.9). Since the random-step world is broken into five equally-sized pieces we expect that



(a) (b)

Figure 4.8: Comparison of languages evolved in identity and random-step worlds. Figure (a) shows the best language evolved in the identity world. Figure (b) shows the best language evolved in the random-step world. No important structural differences between these two languages are readily apparent beyond the difference in size, but relating the structure of the languages to the structure of the world (Fig. 4.9) reveals the concordance between language (b) and the random-step world.

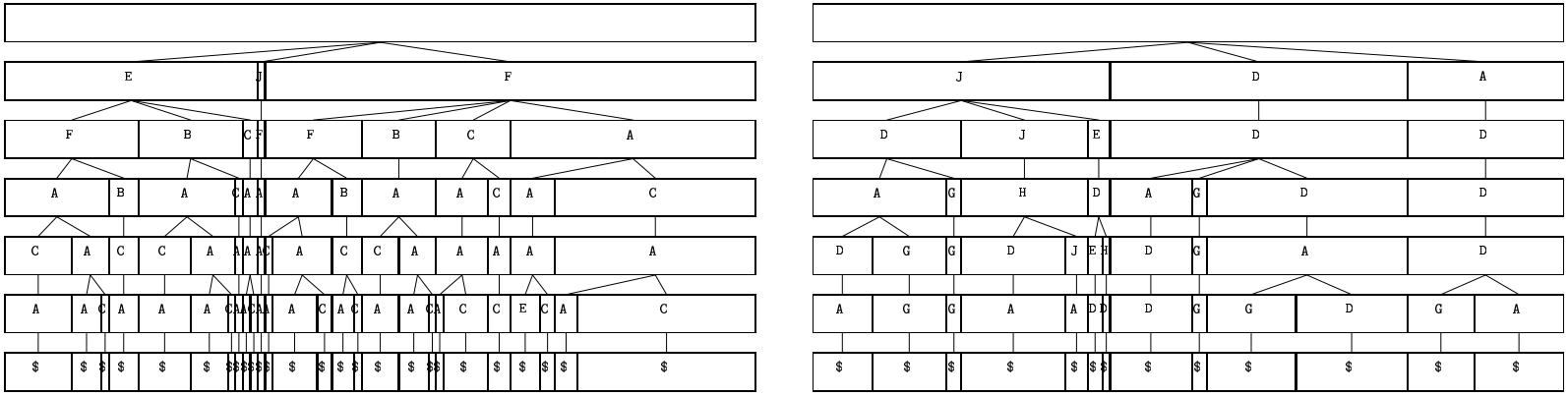
Table 4.1: Learnability of languages in different worlds. Languages were evolved to be learnable in either the identity or random-step worlds, then tested for learnability in the four worlds shown. Each cell summarises the learnability of ten languages, each tested by training 100 new random decoders. The first number in each cell in the table shows the number of languages out of ten that were reliably learnable (i.e., successfully learned by at least 50 of the learners). The number in parentheses is the number of successful learners out of 100, averaged across the ten languages. The two results shown in bold are the cases where the test and evolution conditions were the same. In both cases, learnability is substantially better in 'native' worlds, with seven or eight of the ten languages reliable learnable.

	Test World 10 (100)			
Evolution World	identity	random-step	sine	cubic
identity	**7 (64)**	0 (5)	0 (0)	1 (29)
random-step	2 (24)	**8 (60)**	0 (0)	0 (17)

good languages should use easily distinguished utterances in each range. The language shown in Fig. 4.8(b) demonstrated this effect. The first level of the tree is broken into three regions covered by `J`, `D` and `A`. The `J` symbol covers exactly the first two fifths of the concept space, the `D` covers the next two fifths, and the `A` symbol covers the remaining fifth. Within the region covered by `J`, `JD` covers the first fifth of the (entire) space, and `JJ` and `JE` cover the remainder. Thus, all concepts in the first step are encoded by strings starting with `JD`; those in the second step are encoded by strings starting with `JJ` or `JE`; those in the third and fourth steps are encoded by strings starting with `D` and those in the final step are encoded by strings starting with `A`. The language makes no clear distinction between the third and fourth steps — the border between them falls inside the region encoded by `DDDAG$`. However, these two steps are the least important to differentiate since they have such similar heights (see Fig. 4.7b). Thus, the steps of the world are encoded by the first (and more 'significant' in numeric terms) symbols of each utterance, making the steps easily distinguishable. Conversely, the language that was evolved in the identity world (Fig. 4.9a) has no clear relationship to the random-step world.

4.5 Study 3: Generalisation from a fixed set

In the former two studies in this chapter, decoders were trained on the output of encoders for randomly selected concepts. All concepts were equally likely to be



(a) (b)

Figure 4.9: Comparison of languages evolved in identity and random-step worlds with respect to the concept domain. The languages shown in (a) and (b) correspond with those in Fig. 4.8(a) and (b). These figures show not only the structure of the language, but its relationship to the concept space. The root node shows the entire range of concepts, from 0.0 to 0.99. The width of the box in which each symbol is drawn relates to the region of the concept space covered by that branch of the tree. For example in (a) the smallest concepts are encoded by strings starting with `E`, only one concept is encoded by a string starting with `J`, and almost two thirds of the concepts are encoded by strings starting with `F`.

chosen. In some cases an unfortunate choice of examples made learning extremely difficult, for example, if all of the examples were chosen from the same region of the concept space. In contrast, it seems likely that there is much commonality between the learning environments of human children, and that this environment is in some sense semantically constrained, at least in a statistical sense. This suggestion is different to that of Elman (1993) where the constraints are imposed internally in the infant by constraints on the available processing power. Rather, the suggestion is that the environment of the human infant is shaped by external, non-linguistic processes that produce a predictably structured learning environment.

The third study considered whether languages could adapt to be learnable from a specific set of concepts. The results of study 2 (§4.4) suggested that decoders had difficulty generalising to the sine function. To provide a challenging task, the systems in this study were all evolved with the sine generalisation task. Two sets of simulations were performed (studies 3A and 3B). In the first set of simulations (3A, the *variable environment* condition), ten languages were evolved to be generalisable from ten examples, randomly chosen for each learner. This first set of simulations was essentially a repeat of the simulations of study 2 (§4.4) but with the sine function instead of either the linear or random-step function. In the second set of simulations (3B, the *fixed environment* condition), ten languages were evolved to be generalisable from ten examples. These ten examples were chosen randomly, but were the same for each learner within each of the trials (i.e., one set of concepts was randomly generated for each trial). For convenience, the ten sets of training examples used in study 3B are denoted $\mathcal{A}, \mathcal{B}, \ldots, \mathcal{J}$, and the languages that were evolved for each of these sets $L_{\mathcal{A}}, L_{\mathcal{B}}, \ldots, L_{\mathcal{J}}$. The ten languages evolved in the fixed environment simulations (study 3A) are denoted $L_1, \ldots, L_{10}$.

If languages can indeed boost their learnability by adapting to the learning environment of learners, then it should be expected (a) that languages will be more learnable when the learning environment is stable than when it is not, and (b) that languages will be more learnable in their 'native' environment (i.e., when learners are in the same environment in which the language evolved). To test the first of these assertions, 100 random decoders were trained on each of the ten languages from the variable environment (again using different training examples for each learner) as well as on each of the ten languages from the fixed environment (using the same training examples as during evolution). The learnability of the two sets of languages were then compared.

To test whether languages were particularly suited to being learned from the examples used in the evolution phase, 100 decoders were trained on each of the ten languages ($L_{\mathcal{A}}, \ldots, L_{\mathcal{J}}$) using the example sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ (these sets were chosen arbitrarily). The results from these tests are then compared with the results collected from study 3B, where 100 decoders were trained on the examples used during evolution.

4.5.1 Study 3: Results and discussion

The results from studies 3A (variable environment) and 3B (fixed environment) clearly demonstrate that, on average, a learner is more likely to be successful in the fixed environment than in the variable environment. Languages in the variable environment condition (3A) were successfully acquired by only 2.8 learners on average, compared with 28.3 learners for the fixed environment condition (3B). However, even in the fixed environment a substantial proportion of languages were not learned by any learners. Inspection of the sets of training examples revealed that sets $\mathcal{F}$ and $\mathcal{I}$ were unrepresentative of the concept space. For example, neither $\mathcal{F}$ nor $\mathcal{I}$ contain any examples between 0.3 and 0.6, the region of most rapid change in the sine function. Similarly, training sets $\mathcal{G}$ and $\mathcal{J}$ fail to cover sizeable regions of the concept space where the gradient of the sine function is large. Set $\mathcal{B}$ has no obvious challenging properties. These observations indicated that the language evolution process was incapable of overcoming the challenges imposed by difficult training sets. The learnability of the languages produced from both the fixed environment trials and variable environment trials is shown in Table 4.2.

The results presented in Table 4.2, can not give an unambiguous indication of whether languages can adapt to be learnable from a specific set of examples. Certainly the results show that the fixed environment trials outperformed the variable environment trials on average. However, the results may be a consequence of the training set alone rather than the interaction between training set and language evolution (e.g., $L_{\mathcal{E}}$ may be a particularly well chosen set of examples). A clearer picture of the interaction between training set and language evolution should be given by the results of the second set of tests.

It was expected that languages would be considerably easier to learn from the set of examples from which the language was evolved to be learned. Such a result would follow from the adaptation of the language to the training examples, and would manifest itself by having learnability from the 'native' training set substantially

Table 4.2: Learnability of languages evolved in study 3 for both the variable learning environment (study 3A) and the fixed learning environment (study 3B). The table shows the number of successful learners (to the 1.0 criterion used previously) out of 100 trained for each of the evolved languages. Although the average success rate is considerably higher for the fixed environment, there are a comparable number of languages that are not learned by any learners. Note that the language labels are arbitrary, so that L_1 has no relationship with L_A.

Variable Environment (3A)		Fixed Environment (3B)	
Language	Successful learners (of 100)	Language	Successful learners (of 100)
L_1	4	$L_{\mathcal{A}}$	60
L_2	0	$L_{\mathcal{B}}$	0
L_3	0	$L_{\mathcal{C}}$	49
L_4	0	$L_{\mathcal{D}}$	12
L_5	13	$L_{\mathcal{E}}$	98
L_6	0	$L_{\mathcal{F}}$	0
L_7	0	$L_{\mathcal{G}}$	0
L_8	3	$L_{\mathcal{H}}$	54
L_9	6	$L_{\mathcal{I}}$	0
L_{10}	2	$L_{\mathcal{J}}$	1

better than learnability from a 'foreign' training set. The results are not so clear.

For languages $L_{\mathcal{A}}$, $L_{\mathcal{C}}$ and $L_{\mathcal{E}}$, the predicted pattern of results occurs, and for languages $L_{\mathcal{B}}$, $L_{\mathcal{F}}$, $L_{\mathcal{G}}$ and $L_{\mathcal{I}}$ where all learners failed to reach the criterion, the results provide no evidence either way. However, for languages $L_{\mathcal{D}}$, $L_{\mathcal{H}}$ and $L_{\mathcal{J}}$ the results are contrary to what was expected. For each of these three anomalous languages, learnability is better on training set $\mathcal{C}$ than on the set for which they were evolved. Table 4.3 documents all of the results.

There appear to be two general effects. The first is that some sets of training examples are easier to learn from than others, independent of the language (e.g., set $\mathcal{C}$). The second effect is the expected one, namely that the language does evolve to become learnable from a specific training set (e.g., $L_{\mathcal{A}}$ and $L_{\mathcal{E}}$). However, the results are insufficiently clear to permit any strong conclusions to be drawn along these lines. The notion that languages may exploit consistent learning environments to boost generalisability remains an interesting prospect for future work. A weaker conclusion that may be drawn from the results of this section is simply that the choice of training examples plays an important role. Indeed, one of the main benefits of the environment in which human infants acquire language may simply be that it ensures that the distribution of examples is not pathological.

Table 4.3: Learnability of languages from various training sets. The learnability of each of the ten languages evolved with a fixed learning environment was tested on three alternate training sets, $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. Each cell in the table shows the number of successful learners (to the 1.0 criterion used previously) out of 100 trained for the given language and set of training examples. These results may be compared with the learnability of the language from its 'native' training set, shown in column * (also shown in part 3B of Table 4.2). Note that each of the pairs marked ^{a,b,c} represents the same result.

	Number of successful learners out of 100			
	Training set during testing			
Language	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	*
$L_{\mathcal{A}}$	60^a	21	0	60^a
$L_{\mathcal{B}}$	0	0^b	0	0^b
$L_{\mathcal{C}}$	2	4	49^c	49^c
$L_{\mathcal{D}}$	0	1	44	12
$L_{\mathcal{E}}$	0	0	31	98
$L_{\mathcal{F}}$	0	0	0	0
$L_{\mathcal{G}}$	0	0	0	0
$L_{\mathcal{H}}$	0	0	84	54
$L_{\mathcal{I}}$	0	0	0	0
$L_{\mathcal{J}}$	0	39	98	1

4.6 Discussion: External factors in generalisation

The first study showed that a language could be learned from five strings by a recurrent network. Generalisation performance included correct decoding of novel branches as well as symbols in novel positions (Fig. 4.2). The second study highlighted how a language could be evolved to facilitate different forms of generalisation in the decoder. The third and final study aimed to demonstrate that languages could also be tailored to be generalised from a specific set of examples. In this study, all languages were preferentially learned from either their native training set or set $\mathcal{C}$. The results were not sufficiently clear to be able to draw any strong conclusions, but they were suggestive that some relationship may form between the language and its training set.

The three series of simulations modified the language environment of the decoder in different ways: (1) the relationship between utterances and meaning; (2) the type of generalisation required from the decoder by the external environment; and (3) the particular utterances and meanings to which a learner was exposed. In each case, the language environment of the learner was sculpted to exploit the biases present in

the learner. Batali's (1994) method of providing learners with an additional bias in the form of initial weights was also likely to have been effective in assisting learning. However, the purpose of the simulations in this chapter was to investigate how external factors could assist in simplifying learning, rather than the phylogenetic evolution of language-specific competences.

> "The key to understanding language learnability does not lie in the richly social context of language training, nor in the incredibly prescient guesses of young language learners; rather, it lies in a process that seems otherwise far remote from the microcosm of toddlers and caretakers — language change. Although the rate of social evolutionary change in learning structure appears unchanging compared with the time it takes a child to develop language abilities, this process is crucial to understanding how the child can learn a language that on the surface appears impossibly complex and poorly taught." (Deacon, 1997, p115).

This chapter has considered the ways in which languages may exploit the characteristics of their learners to boost their generalisability. The results suggest that the requirement of an innate language competence may be pushed back — that the biases of general-purpose learning mechanisms can be exploited by judicious choice of language. In all of the simulations in this chapter, enhancement of language learnability is achieved through changes to the learner's environment, without resorting to the addition of biases to the language acquisition device.

The types of languages that were evolved in this chapter are unlike either human languages or the artificial languages that emerged from the simulation work of Kirby (1999b) and Batali (1998). This difference in languages is likely due to differences in the semantic domains, utterance domains, and in the learners themselves. Most significantly, the languages that emerged in this chapter were not compositional in nature: it was not possible to draw a direct correspondence between components of an utterance and components of the meaning. These studies thus provide evidence that compositionality is not the only viable form of linguistic structure, and that compositional generalisation is not a necessary condition for the emergence of language-like communication systems. While compositionality is a dominant feature of *human* languages, the results suggest that there are viable alternatives. Thus, models that strive to understand the *general principles* behind the emergence of language-like communication system should not, like Kirby (2000), utilise a learner that assumes compositionality.

Chapter 5

Emergence of Language in a Population

The studies presented in the previous two chapters have considered the case of a single sender communicating with a single receiver. In these studies, the sender evolved to produce a learnable language that enabled accurate communication. The language that emerged was then used to teach new receivers, yielding a one-to-many relationship between senders and receivers. This simulation design captures an important building block in a communication system but fails to model many other factors. Two features of the model in particular are easily recognisable as misrepresentations of a realistic communication system: the lack of a population of speakers and an artificial distinction between senders and receivers.

In the model that was considered in the previous chapters, there was only ever one sender in any given generation. The language of the system was thus defined by the language of that individual. In more realistic communication systems there are many senders. Each of these senders may possibly have its own idiosyncratic language that differs slightly from that spoken by the remainder of the population. While there may be no canonical language, there is nevertheless broad agreement across the population. Such agreement is obviously advantageous to a population; if individuals agree on a uniform language, then once an individual acquires that language, it can communicate with all members of the population.

One problem for a population is *how* to reach consensus on a language, that is, how the individuals' languages converge on a standard form. As noted in §2.2, population convergence is often studied in the context of signalling systems. However, there are some significant differences between language systems and signalling sys-

tems that seem important for convergence. Primary amongst these differences is the need to co-ordinate the use of linguistic structures. In signalling systems the signals are atomic, whereas for language a population must agree not only on what a given signal means, but also on the meaning of a *composition* of symbols. Furthermore, learners must generalise this structure from a limited subset of examples.

The processing constraints of a language's users may determine some of its properties: the population agrees on a particular protocol because it is the simplest form to process. In this case, consensus is 'built-in' to the users. For example, in Chapter 3, encoders were strongly biased towards MSB first languages, so it would be easy to establish an MSB first language in a population of encoders. Other properties of a language may be determined arbitrarily, agreed on only for the sake of convention. For example, in Chapter 3 there was no reason why `0` should have been preferred over `1` when indicating small values. Indeed, many systems produced languages with 'negated' semantics or other exotic forms (§3.4.2), an expected outcome given the use of arbitrary symbols. The issue of agreement on symbols is essentially the problem of signalling systems, and that literature is informative as to how agreement can be reached (Oliphant, 1999, for example). These two examples (MSB versus LSB and `0` versus `1`) represent opposite extremes; one being entirely constrained by processing limitations, the other being entirely arbitrary. In reality, the situation is far more complex. Most properties fall somewhere between predetermination and arbitrariness and can not, in general, be determined independently of the other properties of the language.

In the traditional generative grammar approach, Universal Grammar (UG) provides strong, innate constraints on linguistic form, thus solving much of the consensus problem. However, English and Japanese are very different languages so there remains some degree of flexibility. The flexibility goes beyond the simple surface features of language, such as which word to use for a given meaning (analogous to the `0/1` case above), and extends into syntactic structures. An alternative explanation that has been considered throughout this thesis is that the dynamics of linguistic interaction, when coupled with learning biases much weaker than the innate constraints of UG, may be sufficient for establishing syntactic conventions. Previous chapters have established that a language can be 'designed' to be easily acquired by such a learner. This chapter considers the conditions under which a *population* of such learners can reach consensus on an appropriate language, that is, the conditions under which the population converges on an easily acquirable language.

In Chapter 3 an explicit distinction was made between the processes of sending and receiving. This distinction was made to demonstrate a point: that language may need to adapt to the different constraints and biases inherent in the sending and receiving tasks. A more likely scenario is one in which the two processes are related. For example, in humans the ability to produce comprehensible utterances comes (at least partly) from listening to others. That is, the knowledge of what to send is based on the receiving process: information is shared between the sending and receiving 'modules.' At this point it is unclear how to best model the dependencies between sending and receiving. (Indeed, it is unclear exactly what the dependencies are.) The complete separation of an individual's sending and receiving behaviours seems inappropriate, as does the assumption that the functions must be inverses. In this chapter, the choice is made for pragmatic reasons, as a result of some preliminary simulations (§5.1).

Thus, the simulations presented in this chapter extend the work of the previous chapters in two ways: the inclusion of a population model, and the ability of agents to both send and receive. The aim is to investigate the conditions under which a population of weakly biased learners can converge on a learnable language. These simulations are closely related to Kirby's Iterated Learning Model that was reviewed in §2.2.1.

Kirby (1999b) showed how the space of observed languages might be constrained by a language learning dynamic. As languages are passed from one generation to the next, they are filtered through the learning experience. Importantly, this filter acts as a bottleneck since a language learner can never observe every sentence in the language. Kirby argues that a consequence of this dynamic is a pressure for languages to evolve towards forms that are easy generalisable by learners, and presents some intriguing simulations to demonstrate his point.

Kirby's simulations (like all computational models), considered an idealised system. Consequently, although Kirby showed that a language-learning evolutionary dynamic was sufficient to evolve a learnable language under a particular set of circumstances, the generality of his results is open to debate. For computational models such as Kirby's, it is important to establish the features of the abstraction that lead to the observed results. That is, we should strive to understand which parts of the abstraction are required, those that are superfluous, and those that must be constrained to a critical range of values.

In this chapter we explore Kirby's simulations in greater detail. Kirby credited

his results to a 'learning bottleneck' but did not test this issue directly as he did not consider variations in learners, tasks or parameters. His choice of learner was motivated by the fact that it had been developed as an algorithm for grammar induction, and the choice of semantic domain was constrained so as to have combinatorial structure. The question we consider is whether the learning bottleneck is the primary factor with other kinds of learners and a differently structured semantic domain. Kirby's learning mechanisms looked for common substrings and inferred generalised rules for generating them. We believe that this assumption is unnecessarily strong, and that a weaker assumption can be tested in an alternative framework.

Preliminary simulations revealed some unexpected behaviour that was to play an important role in determining the simulation methodology used in the studies in this chapter. While these simulations can be said to have failed, the manner in which they did so was instructive, reaffirming the notion that languages adapt to be more easily acquired and demonstrating the need for an obverter-like procedure (that is, using the agent's own 'receive' behaviour to determine its 'send' behaviour, as described in §2.2.1). The basic design and outcome of these simulations is briefly presented in §5.1. The perceived failings of this preliminary model motivated the design of the methodology presented in §5.2 where comparisons are made with Kirby's model, with particular regard to the learning model and the differently structured domain. Simulations within this framework were performed varying two parameters: the amount of training data supplied to the learners (the size of the bottleneck), and the size of the population (§5.3). An analysis of why the results vary across changes in these parameters relates the results back to Kirby's work (§5.4). The simulations of §5.5 further explored how Kirby's results depended upon experimental conditions, this time by varying aspects of the learning environment.

5.1 A first attempt: The obverter requirement

A preliminary simulation design considered the case of a population of encoders. These simulations were closely modelled after Kirby's Iterated Learning Model (see §2.2.1, p26). The population was arranged in a ring. In each step of the simulation, one individual was replaced by a new, untrained individual. The new individual was then taught to mimic the language productions of its two neighbours by observing a set of *(meaning, utterance)* pairs. The meaning domain was the same as in the previous chapter (i.e., the unit interval, approximated by 100 points) as were the

encoders (recurrent neural networks). In the simulations of previous chapters, it was not possible to use BPTT for training the encoders since there was no fixed target output (the language was free to vary) and there were no other decoders from which to learn. In these preliminary simulations, the existence of a population of encoders meant that new encoders had a target language (the language of the pre-existing population) and so could be trained with BPTT rather than the hill-climbing method previously used. The alphabet was reduced to a size of four, with the symbols (`A`, `B`, `C` and `D`) being represented by one-hot encoded vectors (i.e., $[1,0,0,0],\ldots,[0,0,0,1]$).

Initially, the members of the population had unrelated languages. The success of the population was measured by the similarity of different encoders' utterances for a given meaning. It was expected that, over time, the languages of the population would converge on a reliably learnable form.

The simulation results confirmed this prediction albeit in a somewhat unexpected way: the populations converged on totally homonymous languages. In these languages, the same utterance is used to express every meaning, thus making it impossible for the receiver to recover the intended meaning. Such a language is not expressive, but the only requirement on the population was for agreement, not expressivity. Furthermore, these types of languages are perhaps the simplest to acquire and are consequently resistant to change through the language transmission dynamic. Thus, the results are what (in retrospect) should have been expected: the population converges to the attractor of the transmission dynamic — the most reliably transmissible form.

While the results supported the hypothesis, they failed to do so in a satisfactory manner. Numerous changes were made to the original simulation design with the aim of encouraging expressivity in the encoders (i.e., the use of different utterances for different meanings). None of these changes resulted in populations that converged on expressive languages. This string of failures suggested the same conclusion drawn by both Batali and Kirby (personal communication): that a system based on mimicry alone is inadequate and that an obverter-like process is necessary. The simulations utilising the obverter procedure are presented in the following sections.

5.2 Methodology

The basic methodology was similar to that of Kirby's (see p26), with the significant difference that we used recurrent neural networks (RNNs) rather than symbolic

grammars to model communicative agents, and also employed a different meaning domain. In Kirby's original simulations, the agents attempted to communicate simple predicates denoting agent, action and patient ("Who did what to whom.") represented as triples (or 3-tuples). These simulations used the same simple semantic domain as in previous chapters, where meanings were represented as points in the unit interval $[0, 1]$, which for simplicity are restricted to 100 values of 0.01 increments (i.e., $0, 0.01, 0.02, \ldots, 0.99$). Similar to Kirby's simulations, agents communicated a meaning by sending a sequence of up to six symbols which were taken from an alphabet of size four. Following standard neural network practice, the symbols are represented as four-dimensional binary vectors $[1, 0, 0, 0]$, $[0, 1, 0, 0]$, $[0, 0, 1, 0]$ and $[0, 0, 0, 1]$, which will be denoted `A`, `B`, `C` and `D` respectively.

The simulations in previous chapters used one RNN for encoding (taking a meaning and producing a sequence of symbols) and a separate RNN for decoding (taking the sequence of symbols and producing a meaning). The present series of simulations used a population of agents with a homogeneous network architecture, capable of both sending and receiving. After the failures noted above, we determined that the most appropriate approach was to use a population of decoder networks and use the same obverter approach to message production used by Batali (1998). Thus, the decoders of the previous chapters were given the ability to also act as encoders.[1]

To decode an utterance, the activations of the network were reset to zero and the sequence of symbols was propagated through the weights of the network. Once the entire sequence had been propagated, the decoded meaning could be read off the output unit. Producing an utterance involved the obverter procedure — a network attempted to produce the utterance which, if the network were to hear, would be understood to correspond to the desired meaning. That is, if we consider an agent as a mapping from utterances to meanings $A : U \rightarrow M$, then the obverter procedure tries to produce utterances by approximating the inverse of its own comprehension function A^{-1}. The mechanics of the process are the same as those used by Batali (1998) and are best described with a hypothetical example.

Suppose that an agent wants to communicate the meaning 0.63. To do so, the agent must find the utterance which it understands to mean 0.63 (or as close to it as possible). The first step is to reinitialise the activations of the network to zero. The

[1] In this study the decoders were (first-order) RNNs having four input units (corresponding to the vectors that form the symbols available to the language), five hidden units, and a single output unit (corresponding to the one-dimensional 'meaning'. These units were connected in a simple recurrent network architecture (Elman, 1990).

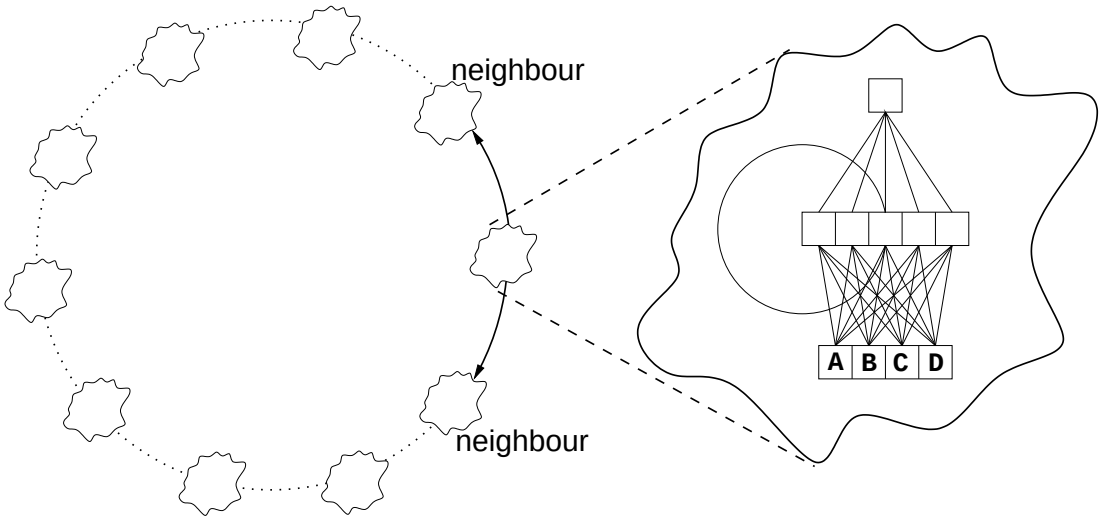


Figure 5.1: A population of communicating agents. Each agent was modelled by a simple recurrent network and could communicate with two neighbours so that the population formed a ring.

sequence of symbols is then determined by an iterative process. Four copies of the network are made, one for each different symbol. Each of the four different symbol vectors is then propagated through its corresponding network, and the outputs are examined. Suppose that the `A` network output is 0.4, `B` is 0.9, `C` is 0.1 and `D` is 0.2. Since `A` produced the output closest to the target meaning (0.63), `A` is taken to be the first symbol of the utterance. Four copies of the `A` network are then made (since `A` was the winning symbol) and the four symbols are propagated through those networks. This step produces meanings for `AA`, `AB`, `AC` and `AD`. Whichever of these four sub-utterances produces the meaning closest to the target is then used to spawn a further four networks. The process repeats until either adding symbols fails to improve the output of the network (for example if `A` was better than `AA`, `AB`, `AC` and `AD`) or the maximum number of six symbols is reached. This final string is then communicated to the other agent(s).

The population of networks was arranged in a ring so that each individual had two neighbours. The following sequence of events closely follows Kirby (2000), and was repeated 2500 times. Each cycle of this algorithm is called a *generation*, even though only one member of the population changes. Thus, 2500 individuals are trained in total. The basic organisation of the population is depicted in Fig. 5.1.

1. Replace a randomly chosen network by a new network with small random

weights taken from a uniform distribution between -0.3 and 0.3.

2. Create a set of *(meaning, utterance)* training examples by encoding a set of randomly chosen concepts with a neighbouring agent. The training set contains utterances produced by both neighbours.

3. Train the new network on the training corpora. The network is trained with BPTT using a learning rate of 0.01 and a momentum term of 0.9 to produce the appropriate meaning upon presentation of an utterance. The entire training corpus is presented to the network 1000 times.

4. Evaluate the *communicative accuracy* of the population in the following way. Every combination of sender and receiver, regardless of location, attempts to communicate the 100 meanings. The squared communicative error for each meaning is summed giving a communicative error score for each (sender, receiver) pair. These scores are then averaged, giving a measure of the average communicative error for the population.

5. Return to step (1).

Two parameters of the simulations were varied — the size of the training corpus and the size of the population — with three variations of these parameters. In the first variation we used a population of size ten and a training corpus of size ten. The second variation increased the size of the training corpus to twenty while keeping the population size at ten. The third variation increased the size of the population to twenty while using the larger training corpus size of twenty examples. This set of simulations is denoted as series 1 and the three combinations of parameter settings as studies 1A (small population, small corpora), 1B (small population, large corpora) and 1C (large population, large corpora). Importantly, the size of the training corpus was chosen to always be significantly less than the size of the full meaning set. Consequently, networks were required to generalise well beyond the examples in the training corpus to communicate about the full set of meanings.

5.2.1 Putting it all together

This section briefly describes what happened during a typical run. The initial population of networks were untrained and generally produced uninteresting languages. Networks were unable to produce enough unique utterances to differentiate each

meaning. Typically, networks were only able to produce three or four different strings which were reused for many of the 100 meanings. In almost all cases each unique utterance was used for a single contiguous range of meanings. For example, a network may have sent `DDDD` for meanings with values between 0.00 and 0.35, `DDBD` for meanings with values between 0.36 and 0.65 and `DBBB` for meanings with values from 0.66 to 0.99. Furthermore, the agents in the population disagreed on which utterance corresponded to a given meaning. The average communicative accuracy was consequently very poor and agents had little success even in understanding their own utterances. (The degree to which an agent comprehended its own utterances could be tested by taking two copies of the agent, one of which acted as sender, the other as receiver, and measuring their communicative error.)

One of the agents was then replaced with a new individual. The new individual was trained on a set of examples produced by its two neighbours. Since the output of the two neighbours was unrelated, the training data for the new network was likely to be a confusing blend. After training, the new network shared some characteristics of the languages produced by its neighbours and was usually able to understand its own utterances. The communicative accuracy of the newly trained network was typically better than the remainder of the population.

After several agents had been replaced and new ones trained, contiguous sections of the population began to have reasonably high agreement on which utterances to use for which meanings. The consistency was never perfect, but networks did tend towards using similar strings for a given meaning. Often, one contiguous subset of the population would use one convention for a region of the meaning space, while the remainder of the population would use a different convention. For example agents one to five may use `AAAB` to communicate 0.50 while agents six to ten use `DDDC` to communicate the same meaning. At this stage, the vocabulary of the agents expanded to around twenty unique utterances. That is, agents were capable of differentiating twenty regions of the meaning space where initially they were able to differentiate only three or four. From this point onwards, the course of the simulation was dependent on the choice of parameters. The next section elaborates on this point.

5.3 Base results: Series 1

For each of the three combinations of population size and training data parameters, three separate runs of the simulation were performed with different seeds of the random number generator producing different sets of initial weights and different choices of training examples. In all cases, simulations performed under the same parameters yielded qualitatively and quantitatively similar results. The results presented here are based on the communicative accuracy of the populations, averaged across the three trials performed for each set of simulation parameters. The communicative error between a sender and a receiver was determined by the squared error between the meaning intended by the sender and the meaning as understood by the receiver, summed across the 100 possible meanings. The communicative error for the population as a whole was taken to be the average communicative error for every possible combination of sender and receiver. Following on from the previous chapter, a communicative error score of one or less is taken to be an acceptable level of communicative accuracy.

With a small population size and with small training corpora (study 1A), the populations always failed to reach consensus on a language, as shown in Fig. 5.2. After a brief initial period where communicative error dropped quickly, the error increased again. Throughout the course of a run, the communicative accuracy of the population continued to oscillate, and even during the better periods, the populations failed to communicate with an acceptable degree of error. During the initial improvement in accuracy and during subsequent periods of good performance, individual's languages showed a reasonable level of agreement with some other members of the population, and there were easily distinguishable 'families' of languages. The populations that are responsible for the periods of high error show little coherence. Although small subsets of the population (two or three individuals) may use languages that are somewhat similar, there is no consensus amongst the population at large.

Keeping the same population size as for the previous study while increasing the amount of training data presented to new agents (study 1B) significantly improved performance (see Fig. 5.3). There was a rapid initial convergence as the population reached consensus on a language. The languages produced across the population were not identical, however they were sufficiently similar for accurate communication. While the performance of the population remained on average quite good,

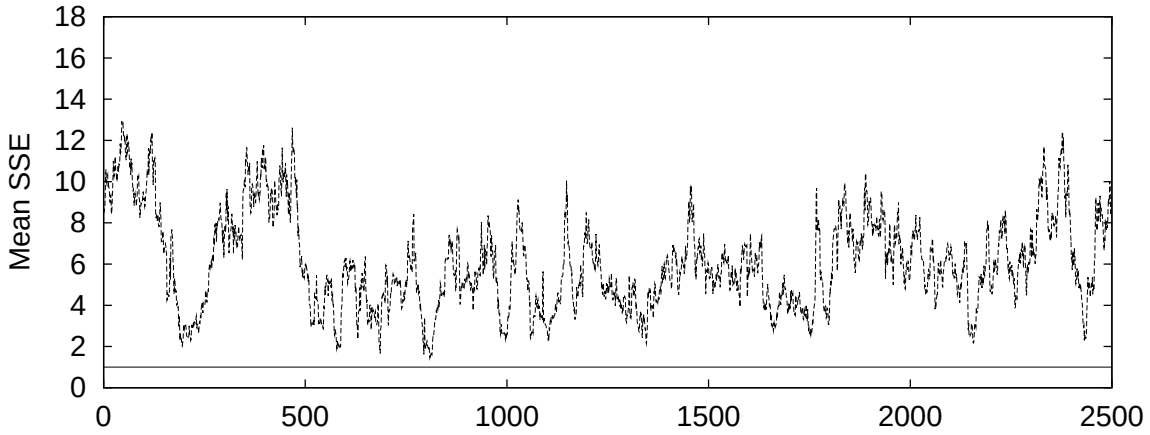


Figure 5.2: Communicative error over time for a population of size ten, using ten examples to train new individuals (study 1A). With these parameters, the population failed to converge on an acceptable language.

there were several transient increases in error. During these periods, part of the population used a significantly different language, where the population agreed on some regions of the meaning space but not on others. Interestingly, the populations on either side of these transient failures may have used languages that were different. That is, following the 'corruption' of the language, the population sometimes reconverged on a different language to that used previously.

Increasing the population size (study 1C) significantly slowed the rate of change of the population (see Fig. 5.4). With the larger population size there was a prolonged period before convergence to an acceptable level of agreement. Indeed, for an initial period the communicative error of the population was substantially higher than at the start. In this region the utterances used by some agents for meanings close to zero were the same as those that other agents use for meanings close to one, and vice versa, giving a worse-than-chance error when they attempted to communicate with one another. Furthermore, under these conditions the population remained unstable in the same way as the case above. Running the simulation for more that 2500 generations revealed that after the population converged, the same increases in error occur. Moreover, the periods of increased error were of greater duration than those observed in the smaller populations.

A representative example of the types of languages found in a population is shown in Table 5.1.

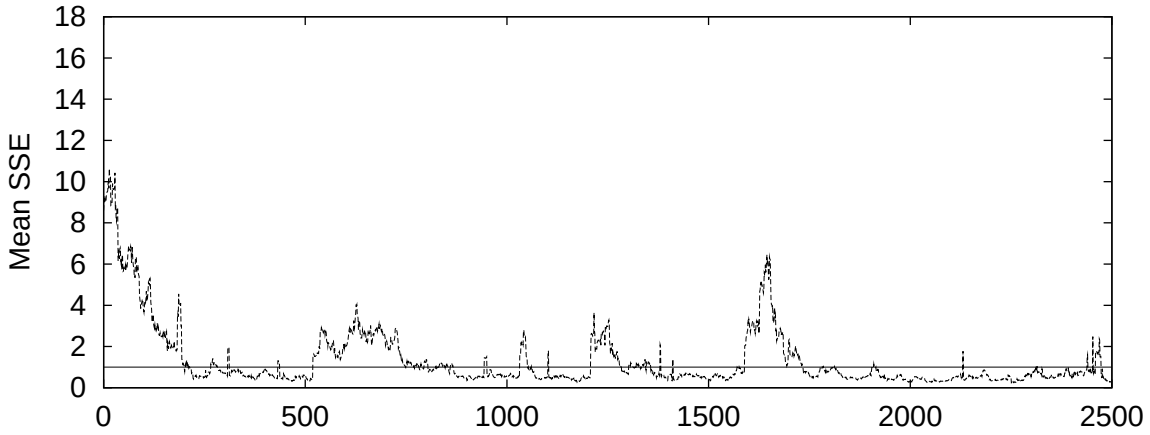


Figure 5.3: Communicative error over time for a population of size ten, using twenty examples to train new individuals (study 1B). Although the population converged to a good language, there were several periods of high error during which two competing languages appeared. In these situations the original language could be replaced by a new variant.

5.4 Analysis of base results

From observing the change in the languages of the population over time we have been able to conclude that much of the behaviour shown in Figs. 5.2, 5.3 and 5.4, and the differences between the study conditions, can be attributed to one cause. Namely, that if a learner failed to acquire the language of its neighbours, then nothing prevented that individual teaching its poorly formed language to subsequent learners. The most significant factor in the failure of an individual to learn was the data presented to the learner. If the ten or twenty training examples were chosen poorly (for example, if they were all less than 0.5), it was much harder for the learner to successfully generalise to the remainder of the space. Utterances for similar meanings tend to be similar so if an agent knew the utterance associated with a meaning such as 0.78 it was more likely to be able to guess the meaning of the utterance associated with 0.75 than it was to guess the meaning of the utterance associated with 0.10.

As the number of training examples increased, the probability of an inadequate sampling of the space diminished. Hence, the population shown in Fig. 5.2 which used ten training examples was far less stable than the population shown in Fig. 5.3 which used twenty training examples. Other factors, such as the initial weights of

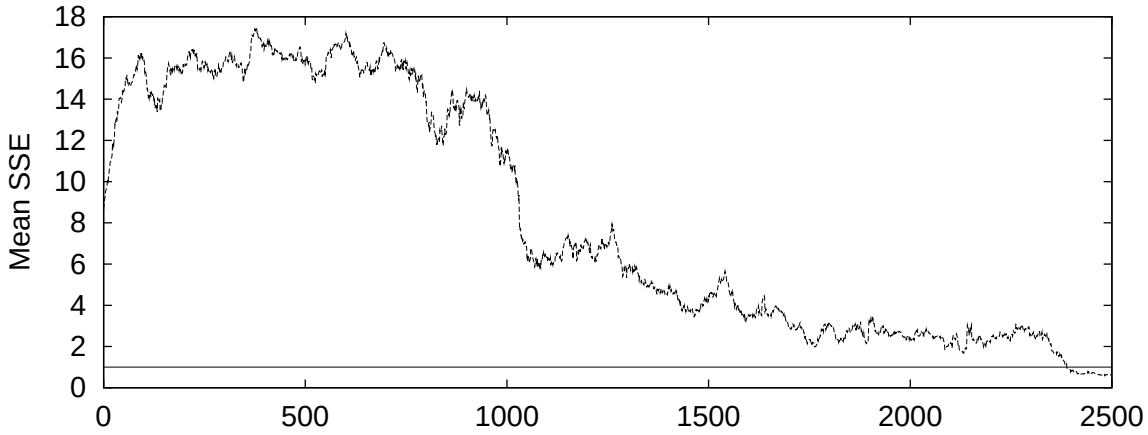


Figure 5.4: Communicative error over time for a population of size twenty, using twenty examples to train new individuals (study 1C). The population behaved similarly to that in Fig. 5.3 but on a much slower time-scale. If the population was allowed to run beyond the 2500 generations shown here, similar intrusions of rogue languages caused intermittent periods of high error.

the learner may have also caused learning failures. However, further simulations (§5.5.1) indicate that the initial weights did not play as significant a role as the distribution of training data.

The differences in time to convergence between Figs. 5.3 and 5.4 can be attributed to greater propagation delays associated with the increase in population size. With a population of size ten, individuals were at most five neighbours away from any other individual. Consequently, the speed with which a change in a language could propagate through the entire population was much greater than with the larger population size (twenty). Once a population formed two (or more) distinct languages it also took a greater time before one came to dominate. Assuming that the languages were equally learnable, one comes to dominate only through providing a disproportionate number of examples in the training corpora of new individuals. Since there was random selection of which neighbour provided a training example, language dispersal involves a degree of chance. An increase in population size increased the size of the region that had to be 'conquered', slowing the dispersal process.

Table 5.1: The utterances used by a neighbourhood of a population for a subset of the meaning space. This small sample shows two competing language forms. Where the first three agents used strings beginning with B for meanings with low numerical values, the other three agents used strings beginning with D. Note that agent 4 showed some similarities to both families. This example also demonstrates that even within one language 'family' there was significant variability.

Concept	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Agent 6
0.00	BBBBBB	BBBBBB	BBBBBB	DDDDDD	DDDDDD	DDDDDD
0.01	BBBBBB	BBBB	BBBBBB	DDDDDD	DDDDDD	DDDDDD
0.02	BBBBBB	BBB	BBB	DDDDD	DDDDDD	DDDDDD
0.03	BBBBBB	BB	BBB	DDDD	DDDDD	DDDDDD
0.04	BBBBBB	BB	BB	DDDB	DDDD	DDDDDD
0.05	BBBBBB	BB	BB	DDD	DDD	DDDDDD
0.06	BBBBBB	B	BB	DDD	DDD	DDDDDD
0.07	BBBBBB	B	B	DDD	DD	DDDDDD
0.08	BBBB	B	B	DDB	DD	DDDDDD
0.09	BBB	B	B	DDB	DD	DDDDDD
0.10	BBB	B	B	DDB	DD	DDDDDD
0.11	BB	B	B	DD	DD	DDDDDD
0.12	BB	B	B	DD	D	DDD
0.13	BB	BDB	B	DD	D	DDD
0.14	BB	BDB	B	DD	D	DD
0.15	BDD	BDB	B	DD	D	DD
0.16	BDD	BDB	BDB	D	D	DD
0.17	BD	BDB	BDB	D	D	DD
0.18	BD	BD	BD	D	D	DD
0.19	BD	BD	BD	D	D	D
0.20	B	BD	BD	DB	D	D
0.21	B	BD	BD	DB	D	D

5.5 Varying the learning environment: Series 2, 3 and 4

Just as in Kirby's simulations we have seen the emergence of co-ordinated, structured communication as a result of the dynamics of linguistic transmission. While not all of Kirby's results have been replicated (which we would not expect given the changes made to Kirby's simulation design), we have seen that one of the significant outcomes (structured communication) does replicate with a different learning mechanism and a different semantic domain. We have also see that a successful outcome can be highly dependent on such factors as the size of the population and

the amount of training data available to new individuals. In this section we consider alternative aspects of the learning environment that can influence the outcome of language evolution. The analysis of the first series of simulations indicated that part of the reason why populations could fail to converge was that a single learner with an idiosyncratic language could corrupt future generations. Kirby explicitly sought to simulate language emergence in the absence of selection pressure to explore the power of glossogenetic adaptation alone. Hence, idiosyncrasies could not be eliminated from a language by a mechanism that removed the poorer speakers from the population. Consequently, the three factors that we vary in series 2–4 are chosen for their potential to either prevent learners from failing, or to stop failed learners propagating their half-formed languages.

It is well understood that failures in neural networks to learn a task can often be attributed to the choice of the initial weights (Kolen and Pollack, 1990). Simulation series 2 repeated the simulations of series 1, but instead of generating the initial weights of new individuals randomly, all new individuals started with the *same* weights. Making this change allowed a language to emerge that was learnable from a specific starting point. This technique has proven successful in other work (Tonkes et al., 2000; Batali, 1994).

Another potential cause of learning failure that we have identified is the selection of training data from which new individuals learn. Learners were presented with a set of *(meaning, utterance)* pairs, where the meaning was a value between 0 and 1. If the selection of meanings in the training sample failed to provide sufficient coverage of the full meaning space, then it was much harder for the learner to generalise to unseen examples as they were dissimilar to the previously seen examples. In simulation series 3, rather than training new learners on different, randomly chosen examples, new learners were trained on the same (randomly chosen) meanings.

In series 4, the variation to series 1 was that the 'neighbourhood' assumption was violated. Instead of using neighbours to provide the training data for new individuals, a 'teacher selection' principle was applied. After every time-step, each individual was given a score based on how well it was understood by the rest of the population (i.e., the portion of error that an individual contributed to the overall error, as plotted in Figs. 5.2, 5.3 and 5.4). This score was used to select which networks generated the examples in a training corpus presented to a learner, based on a proportional selection mechanism (the probability of selection was inversely proportional to error). If a network failed to learn the language of its community

then it would be unlikely to be selected to provide examples to train new individuals, thus limiting its impact on future generations.

In summary, the simulations of §5.2 three were repeated under three different conditions:

1. Using the same set of initial weights for each new learner (series 2: fixed weights).
2. Using the same set of meanings to train each new learner (series 3: fixed examples).
3. Choosing the 'best' networks to generate the training examples for the new learners (series 4: teacher selection).

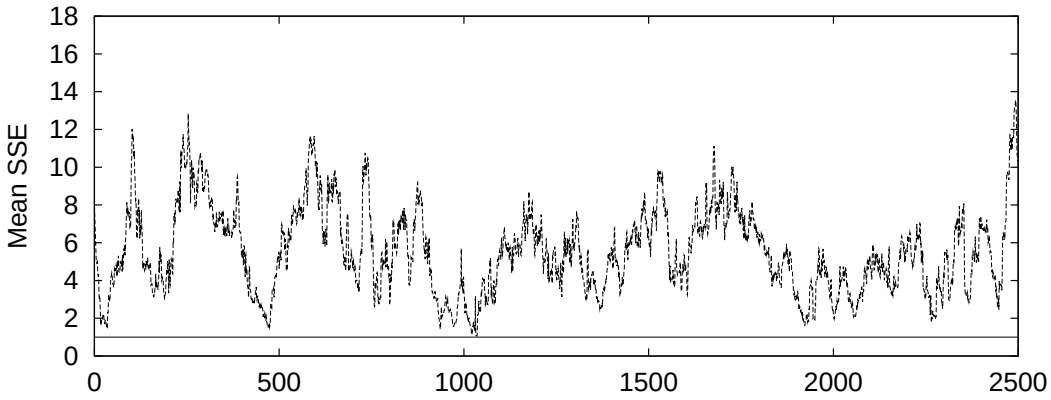
Again, population size and the training corpus size were varied and the simulations from three different random seeds were repeated under each condition (i.e., three repetitions of each of studies 2A, 2B, 2C, etc.).

5.5.1 Results of Varying the Learning Environment

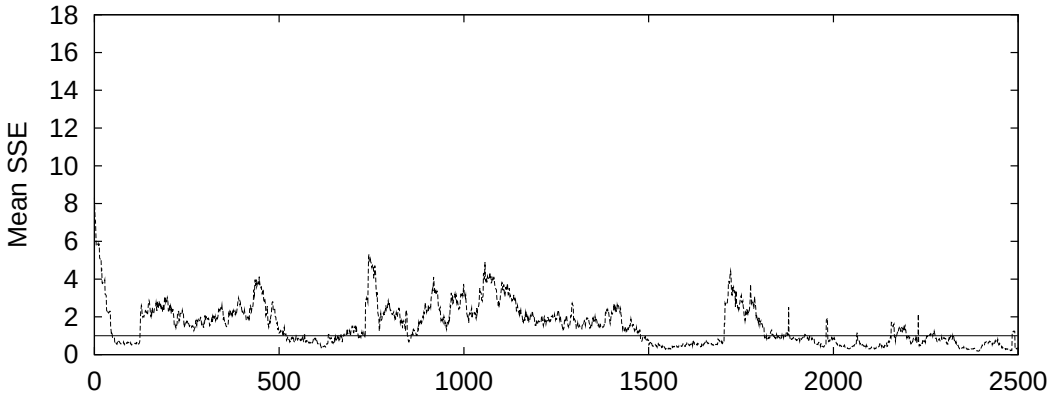
In all cases, the three repetitions of a condition yielded quantitatively similar results. However, across the conditions, the results varied radically. In the 'fixed weights' condition, the populations rapidly achieved reasonably low communicative error (see Fig. 5.5). This effect may be attributable to the fact that all members of the initial population were identical (having the same, untrained weights). However, as in the original series of simulations, the population was unable to maintain this low degree of error and the error fluctuated markedly.

In stark contrast, the populations in the 'fixed examples' condition took longer to converge in each case but showed a remarkable degree of stability (see Fig. 5.6). Although there were some increases in error after the population had apparently converged, the error remained low. Surprisingly, there was no significant difference in the accuracy of the networks when the amount of training data was varied. As before, the C condition resulted in a slower progression towards the general pattern found in the A condition.

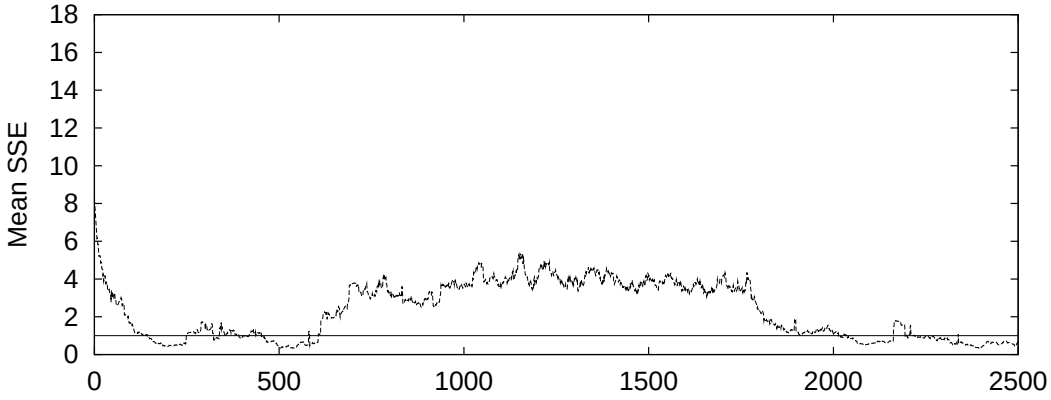
The populations in the 'teacher selection' condition demonstrated yet another pattern of error (see Fig. 5.7). Again, the population rapidly attained a reasonable degree of communicative accuracy (low error). Any increases in error were very



(2A) Small population, less training data.

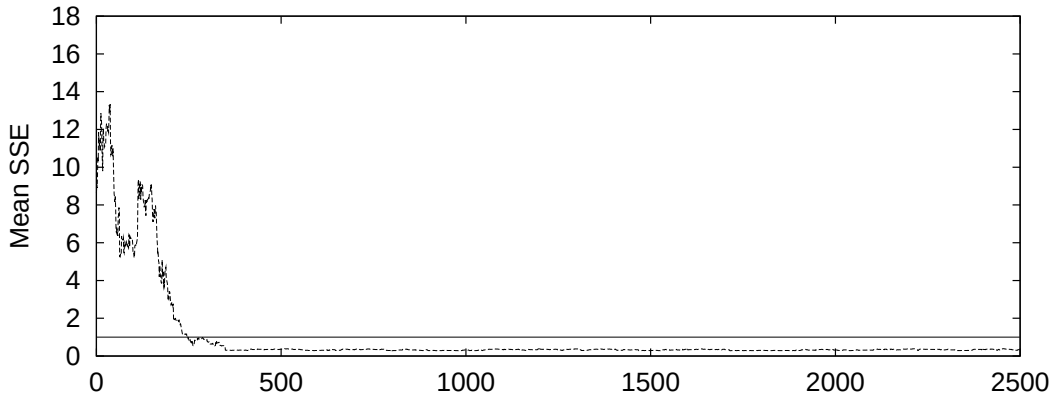


(2B) Small population, more training data.

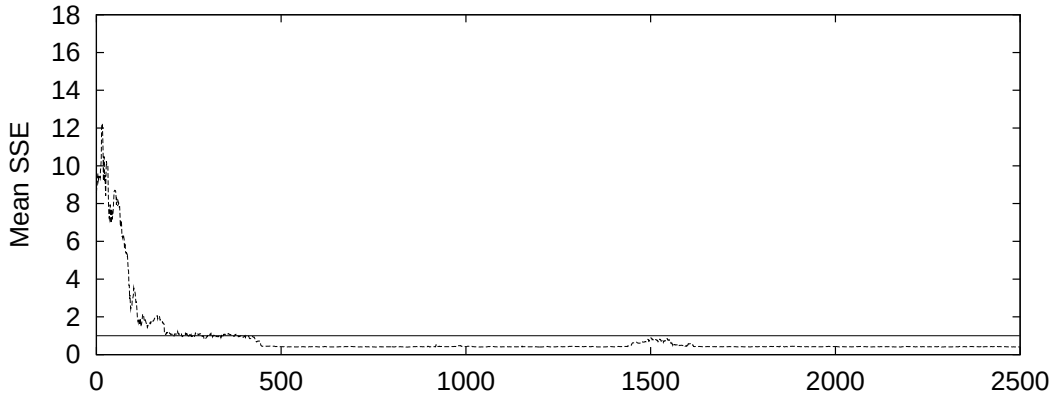


(2C) Large population, more training data.

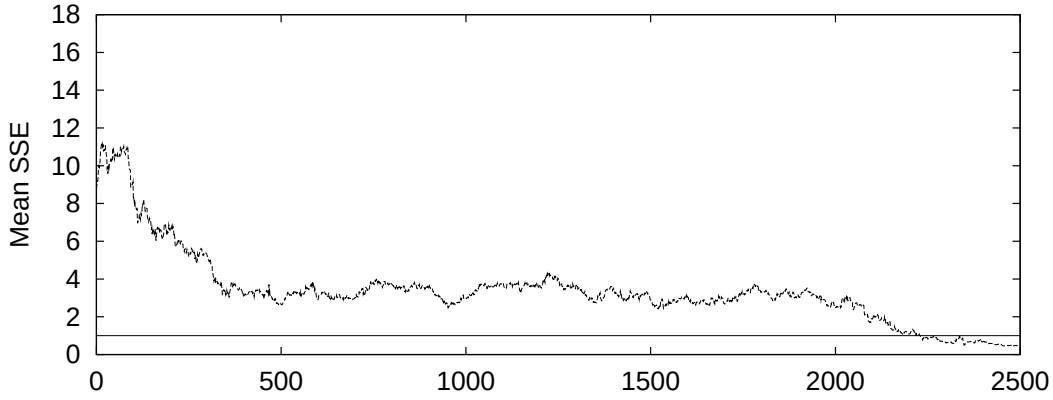
Figure 5.5: Communicative error of populations over time when new individuals always started from the same initial weights (series 2). Since all individuals were originally identical, the population converged quickly. However, as in §5.3 the population frequently departed from an established convention.



(3A) Small population, less training data.



(3B) Small population, more training data.



(3C) Large population, more training data.

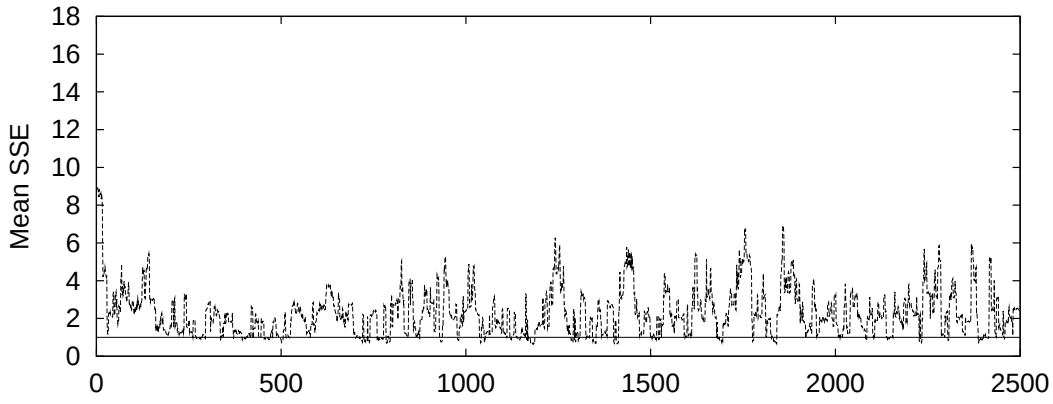
Figure 5.6: Communicative error of populations over time when new individuals were always trained on the same set of meanings (series 3). In all cases, the population was much more stable than its counterpart in the original simulation. Convergence was still slow for larger populations.

short-lived, far more so than in the original simulations. With a small population and a small amount of training data (study 4A) the population was still unstable, but was much better on average than in the original simulations (Fig. 5.2). Even with a larger size, the population very quickly arrived at a point of low error and tended to remain there, despite the occasional increases in error.

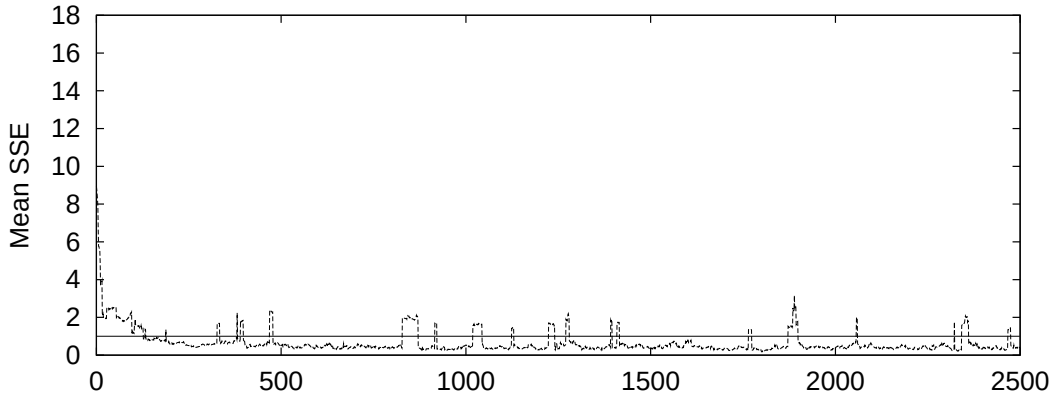
5.5.2 Analysis of Learning Environments

As in §5.3, the changes in the population were analysed by observing the changes in the languages generated by each population. Apart from the initial improvements in communicative accuracy, the results of the 'fixed weights' populations were effectively the same as in the original study, indicating that the choice of initial weights was largely irrelevant. Conversely, the performance of populations in the 'fixed examples' condition suggested that the choice of training data was of vital importance. In this condition, only a single training corpus was generated. The probability that this particular corpus was unrepresentative of the meaning space was small, as it was for networks trained in the original simulations. In the original simulations, 2500 different corpora were generated, one for each learner. The probability that some of these corpora were unrepresentative of the meaning space far exceeds the probability that the single corpus in the later simulations was unrepresentative. If, by chance, the single corpus was chosen poorly, we might expect that the population might never have been successful. The results of the 'fixed weights' and 'fixed examples' simulations lead us to hypothesise that the populations evolved languages to a point where they were reliably learnable regardless of the initial weights of a network, and that only poorly chosen training samples prevented individuals from learning.

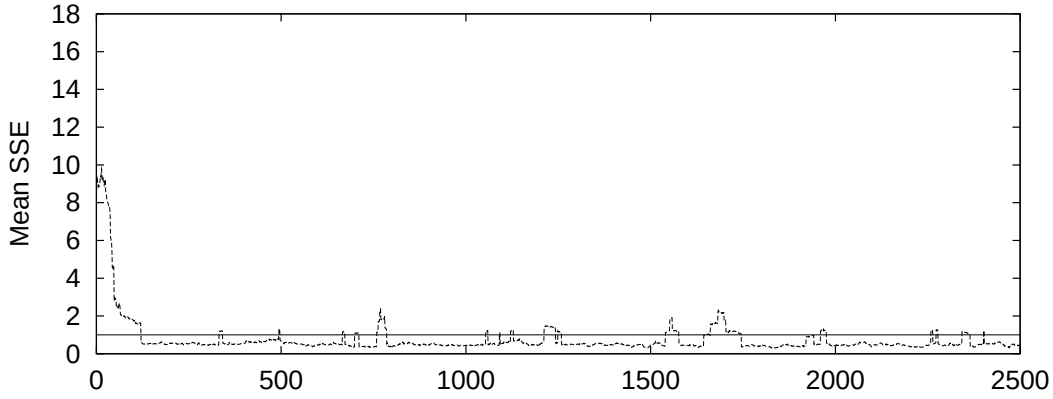
Populations in the 'teacher-selection' condition successfully reduced the influence of rogue learners. The impact can be best seen from the length of time that any population experienced high error. Particularly with a population size of ten and a training corpus of size twenty (study 4B, the middle graph in Fig. 5.7), the length of periods of increased error closely followed the expected lifespan of an individual (ten time-steps on average). This observation suggests that while a rogue learner may have lowered the communicative error of the population, it did not pass its incompatible language to future generations. Communicative accuracy was thus restored once the rogue learner left the population. The effect was much less clear with the smaller training corpus since the probability of multiple successive failed



(4A) Small population, less training data.



(4B) Small population, more training data.



(4C) Large population, more training data.

Figure 5.7: Communicative error of populations over time when new individuals were taught by the better communicators in the population (series 4). Convergence was rapid, even for larger populations. Periods of higher error tended to be transient.

learners was considerably higher. Increases in the number of inconsistent networks in the population increased the probability of further inconsistent networks, hence the instability in this case.

5.6 Discussion

This final section considers what correspondences can be drawn between the framework of these studies and characteristics of human language learners and environments. Simulations of populations of communicating simple recurrent networks showed that in favourable circumstances, languages could emerge in the absence of phylogenetic adaptation (§5.3).

The results demonstrate that one of Kirby's major findings — that a structured communication system can emerge from the dynamics of language transmission — has a generality beyond his original domain. While the kinds of language structures that emerged in our simulations were significantly different to those that emerged from Kirby's simulations, such a result should not be unexpected. The agents employed the most appropriate structures for their respective communication tasks. Given the structure of the languages produced in these studies, the results of the simulations may be used to refute claims that classical compositional syntactic structures are the only viable form of linguistic structure. Thus, human languages exhibit compositional structure not because it is the only valid alternative, but because other constraints on human communicative needs (such as the similarity structure of meanings as represented in the human mind) necessitate compositionality.

The effect of manipulating the two parameters in the simulations — population size and training corpus size — suggests some interesting implications for human languages. The results showed that populations converged on languages regardless of the population size, although time to convergence was slowed by the larger population. Conversely, increasing the size of the training corpus (which can be viewed as increasing a learner's exposure to language, perhaps by increasing the critical period) vastly improved the success of populations. While it is not possible to claim that the same would be true of human populations, the results suggest an interesting hypothesis for the emergence human languages: that a precondition of language emergence is a sufficiently protracted period of learning so that an infant will be exposed to a critical volume of training data.

The modifications to the learning environment made in section §5.5 are also sug-

gestive of the desirable conditions for language emergence. The first modification (fixed weights) may be viewed as analogous to a very weak genetic endowment of linguistic knowledge. This modification proved unsuccessful at improving the communicative accuracy of the population. In the second modification (fixed examples), the learning environment is consistent for every individual — every learner has the same set of experiences. With this environment, populations were far more successful at accurate communication. It is not outlandish to suggest that for humans, there is some degree of commonality between learning environments, although no two humans will share the exact same set of experiences.

Preventing failed learners from acting as teachers was also effective in maintaining the language of a population, but still required that learners were given sufficient training data. This condition introduced a selection mechanism, something which Kirby deliberately avoided adding. However, in populations where learners can fail, and then corrupt future learners, our simulations show that some kind of selection mechanism is important to maintain population stability. Such a mechanism may be manifested in a real-world situation by the direction of a learner's attention away from speakers with impaired language abilities, or by the social exclusion of such speakers, so that they do not contribute to learners' input to the normal extent.

Although Batali (1998) also used neural networks in his simulations, he did not include any generational component, instead using a static population. In his model, the agents in the population communicated amongst themselves until a consensus was reached. Consequently, after the first round of 'negotiations,' agents were no longer naive about the language of the community, making it difficult to look at changes in the language due to selection pressure for (naive) learnability. Batali also used a different semantic domain and his population lacked any kind of spatial organisation. However, it is interesting to note that Batali's populations were successful in producing basic combinatorial language structures despite the lack of an explicit 'learning bottleneck' — the very mechanism to which Kirby ascribes the success of his simulations. One possible explanation for this disparity is that the learning mechanism itself may provide an implicit bottleneck. One feature of neural networks is their tendency to generalise based on similarity. Consequently, it is much easier for a neural network to learn a regular language than an irregular one; it may even be the case that a neural network will be *unable* to learn some irregular forms. In a series of negotiations, it would thus be expected that the more easily learnable forms (i.e., the regular languages) would persist — networks would compromise

on the easier forms. By contrast, in Kirby's simulations, learners were not able to 'forget' associations between meanings and utterances: once a learner acquired an association, it remained for life. Thus, Kirby's learners lacked this implicit bottleneck since they always succeeded at finding a grammar that was consistent with the training data (assuming that the training data itself was consistent).

To provide a comparison between Kirby's explicit bottleneck, and the hypothesised implicit bottleneck of the neural network learner, we ran a control study which repeated the first series of simulations (described in §5.2), without removing individuals from the population. Instead, an individual was chosen to be given additional (learning) exposure to the language of its neighbours as in Batali's simulations. With small populations and small training corpus sizes, the population quickly reached a communicative error score of around one. The languages of these populations were still unstable, although not to the same extent as the population shown in Fig. 5.2. Increasing the amount of training data received in each round resulted in a much more stable population. Even though populations in this condition periodically disagreed, such events were not as catastrophic as those in Fig. 5.3. With a large population and large training corpora, populations were slow to attain reasonable communicative accuracy, much as in Fig. 5.4, though the initial period of very high error was much shorter.

These results, although they are only preliminary, suggest that Kirby's explicit learning bottleneck may not be necessary. Certainly, they indicate that the role of the bottleneck is not as straightforward as Kirby described. Of course, in the case of human languages there clearly is such a bottleneck between generations of learners. Further work may help to determine whether this bottleneck plays a fundamental role, or is merely incidental to the course of language emergence. What seems plausible is a relationship between the implicit bottleneck of the learning mechanism, and the explicit bottleneck in Kirby's simulations.

The major contribution of this chapter is to broaden our ideas of when structured communication systems emerge (and are stable) and when they do not. These studies suggest that critical factors for the emergence of these systems include the training corpus size, and the method for selecting training data. The chapter also considers the *types* of language structures that emerge from a given situation. Human languages are the only natural example of symbolic structured communications systems that we have. It is difficult to establish the causes for such unique phenomena. Computational models allow us to construct a variety of communication systems

and to explore the conditions under which language-like systems can emerge. By examining the conditions under which language does, and does not emerge, we can explore hypotheses about the significant aspects of the human environment that led to the evolution of human languages. The long-term goal is to deduce the general principles behind the emergence of language and properties of those languages. The work presented in this chapter represents a small step towards that goal.

Chapter 6

Discussion

This thesis has explored a perspective for explaining the origins of linguistic structure that is based on considerations beyond the constraints of the language acquisition device. In particular, the work presented in this thesis has considered the notion that the processes of language acquisition and use create a dynamical system through which linguistic structure emerges (see Fig. 6.1). This thesis has presented simulations that have probed the relationship between features of the transmission dynamic and features of the emergent linguistic structures.

6.1 Summary and review

The motivation for considering the language transmission dynamic was presented in **Chapter 2**. This chapter began by contrasting two approaches to understanding human linguistic abilities: the generative grammar approach and the connectionist (or dynamical) approach. One of the major differences between the two approaches that the chapter highlighted was the extent and nature of innate linguistic knowledge: strong, domain-specific constraints (UG) versus weaker, domain-general learning biases. The adaptation of language to the user was reviewed as an alternative theory for explaining how human infants acquire the language of their community so readily. In this viewpoint, linguistic structures are an emergent property of the dynamics of linguistic transmission which arise in response to the needs of language users and learners. The Evolution of Language community was identified as having studied the problem of human linguistic competence from this perspective.

The latter half of Chapter 2 reviewed previous approaches to modelling the evolution and adaptation of language. This work spans a broad array of fields

and considers a range of linguistic phenomena. Within this broad field, agent-based computational modelling of syntax was identified as the area of interest and two previous models were reviewed in detail: Batali's (1998) Negotiation Model and Kirby's (2000) Iterated Learning Model.

Chapter 3 discussed the issues involved in formulating a model of language adaptation. This discussion was then used to introduce the basic simulation framework that was used in the remainder of the thesis. The proposed framework comprised a semantic domain (the continuum of points between 0 and 1), a language learning and processing mechanism (recurrent neural networks coupled with the backpropagation-through-time algorithm) and a message domain (sequences of symbols encoded as binary vectors). This simulation framework was then used to investigate how linguistic structure might adapt when sender and receiver had different learning biases.

Preliminary simulations considered sender and receiver separately and demonstrated that, for the proposed model, the languages that were most suited to the sender were the *reverse* of those most suited to the receiver. Further simulations considered the combined sender-receiver system under two different conditions. In one condition, messages were reversed between sender and receiver (that is, the receiver received the reversed message of the sender) thus making the same language suited to both sender and receiver. In the other condition, message order was preserved. Analysis of the languages that emerged in these simulations revealed structural differences between the languages produced in the two conditions. The structure of languages in the preserved order condition (where sender and receiver preferred different languages) showed evidence of a compromise between the differing constraints of sender and receiver. The results of these simulations demonstrated a situation in which linguistic structure was determined by the intersection of the sender's (quite strong) constraints with the receiver's (relatively weak) constraints.

The simulations of **Chapter 4** considered the role that the bottleneck of linguistic transmission plays in determining linguistic structure. The initial simulations examined the extent to which emergent linguistic structure could overcome the problems posed by this bottleneck for a naive, domain-general learner. Building on the results of Chapter 3 it was shown that, through judicious selection, linguistic structure could be capable of facilitating a significant degree of generalisation without the need to add specific constraints to the learner. While the learners may have been domain-general they were not unbiased. Hence, the languages could ex-

ploit the inherent biases of the learners to boost their own generalisability. The linguistic structures that emerged were successful because they matched the generalisation characteristics of their learners. These simulations demonstrated that language adaptation could facilitate acquisition by a general-purpose learner despite a particularly constricted learning bottleneck.

Additional simulations in Chapter 4 considered how properties of the bottleneck resulted in different structural features in the emergent languages. Manipulating the aspect of the meaning that the sender was required to communicate showed that the structure of the emergent languages was specifically adapted to the communicative task — that the structure of the languages facilitated particular forms of generalisation. A final set of simulations in this chapter aimed to show that generalisability could be further boosted if aspects of the learning environment were kept constant. That is, the bottleneck was made to be consistent between generations and thus allowed linguistic structure to be generalisable from a specific learning environment. While the results of these simulations were inconclusive they were suggestive of some adaptation taking place.

The single-sender/single-receiver simulations of earlier chapters were extended to a population of agents in **Chapter 5**. This extension enabled the investigation of how the properties of populations influenced the dynamical characteristics of linguistic transmission and thus the emergence of structure. One of the noteworthy problems for a population is reaching consensus on linguistic structure. That is, the requirement for the members of the population to have similar languages that are mutually understood. The first study in this chapter investigated how convergence on a learnable language was affected by (a) population size and (b) training corpus (or bottleneck) size.

The results showed that under suitable conditions the dynamics of linguistic interaction were sufficient to establish a learnable language throughout the population, as in Kirby's (2000) simulations. This result indicates that Kirby's findings regarding the emergence of linguistic structure from the dynamics of linguistic transmission have generality beyond his chosen model. Furthermore, the simulation results suggest that the dynamic is *parameterised* by (at least) the population size and training corpus size. Particularly, that in the Iterated Learning Model

- the bottleneck size has a substantial impact on the likelihood of a successful outcome, with larger training corpora preferable;

- the population size controls the rate at which languages propagate through the population, but has little effect on the likelihood of a successful outcome.

Analysis of the behaviour of populations revealed that the major factor preventing the stable convergence of the population on a language was the occurrence of learners that failed to acquire the language of the population. These learners then passed their 'corrupted' languages on to later learners, corrupting the language of the entire population. Further simulations were performed to investigate mechanisms by which the damage from these failed learners could be minimised. That is, factors that could change the linguistic transmission dynamic so as to lead to a more successful outcome. Three mechanisms were tested: (a) keeping the initial state of the learners constant, (b) keeping the learning environment constant, and (c) using the better speakers in the population to provide the training material for new individuals. Of these approaches, (b) and (c) proved effective at helping the population maintain a uniform language. The constancy of the learning environment and the method for generating training material can thus be seen as important principles guiding the dynamics of linguistic transmission.

6.2 Dynamics of linguistic transmission revisited

In terms of addressing the question of how linguistic structure emerges from the dynamics of linguistic transmission, the simulations of Chapters 3 and 4 are qualitatively different from those of Chapter 5. With respect to Fig. 6.1, the simulations of the earlier chapters considered only a single iteration of the transmission dynamic and applied an ad hoc method (the hill-climbing algorithm) to optimise the language for that single interaction. The results of these simulations can therefore not be taken as evidence of the types of linguistic structures that are likely to emerge, but rather the types of linguistic structures that are more likely to persist if they do emerge (or alternatively, the types of linguistic structures that a language must exhibit to survive).

The contribution made by the simulations of Chapters 3 and 4 lies not in showing the emergence of linguistic structure, but in highlighting the extent to which a specifically chosen language can facilitate acquisition by a general-purpose learner. If, as proposed, the dynamics of linguistic transmission are sufficient to produce desirable languages, then the results suggest that the need for innate, domain-specific constraints on language acquisition can be considerably weakened from those claimed

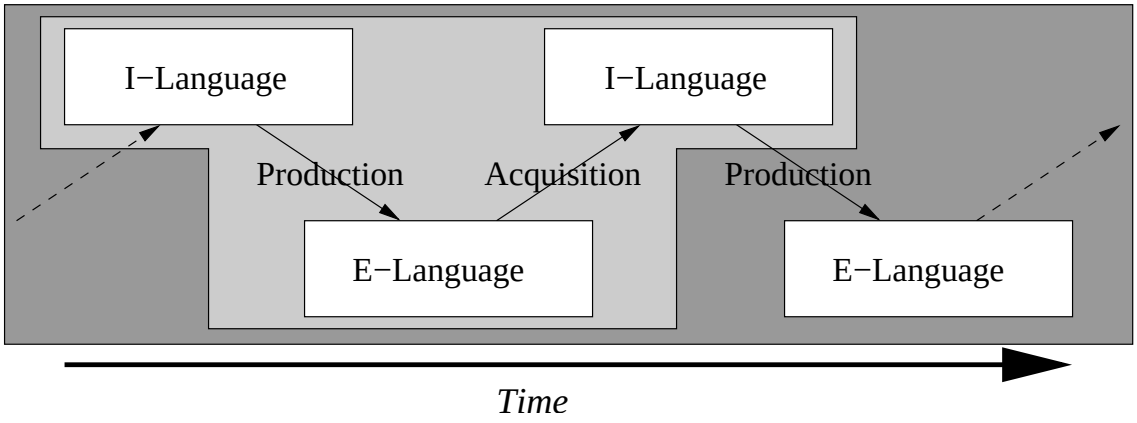


Figure 6.1: The dynamics of language transmission as explored in the thesis. A language dynamic results from repeated cycles of production and acquisition. Chapters 3 and 4 optimised languages by considering only one iteration of the dynamic, shown as lighter shading. Chapter 5 considered the full iterated system, shown as darker shading. (Adapted from Kirby, 2001, Fig. 5, p109; also appearing in this thesis as Fig. 2.2).

by proponents of generative grammar. This demonstration of the ability of language adaptation to facilitate acquisition by a domain-general learner is one of the significant contributions of this thesis.

The simulations of Chapter 5 considered the iterated dynamic of language transmission, rather than the single transmission from speaker to learner studied in the previous chapters. Whereas the earlier chapters investigated the extent to which a language could be synthetically adapted to a general-purpose learner, Chapter 5 was concerned with studying the conditions under which the transmission dynamic could act as a *generator* of learnable languages. Thus, the contribution made by Chapter 5 is in exploring the range of conditions under which a population of general-purpose learners can, through the dynamics of linguistic transmission, generate (and reach consensus on) a learnable language.

6.3 Implications of methodology

The decision to employ a connectionist learner impacted the thesis in terms of both the aspects of language emergence that were explored as well as the basic methodological approach.

- One of the motivating factors for choosing connectionist learners was their general-purpose approach to learning via algorithms such as backpropagation

through time. This thesis has successfully demonstrated that language can be tailored to exploit the biases of such a general-purpose learner, thus facilitating acquisition.

- Compared with Kirby's (2000) symbolic agents, the connectionist agents have different properties with respect to expressivity (using a unique utterance for each meaning). With Kirby's agents there are no constraints in creating arbitrarily complex mappings between utterances and meanings. Indeed, for Kirby's agents it is quite easy to represent (but not necessarily learn) a language that is composed of random associations between utterances and meanings. In contrast, it was difficult to reduce homonymity in the connectionist encoder. The networks naturally mapped similar meanings to similar utterances and it was often difficult for them to discriminate between adjacent meanings. This difference in the biases of the agents resulted in differences in the emergent languages.

- The presence of distinctly different (in fact, opposite) biases between sender and receiver was a direct result of using connectionist models. These biases of recurrent networks were unexpected but allowed the exploration of how language could evolve to mediate conflicting biases.

- Connectionist models, while conceptually simple, are computationally expensive. When coupled with the evolutionary approach explored in this thesis, the computational demands become substantial. Consequently, the studies in this thesis were only able to explore a limited domain so that simulations remained tractable. Even so, significant compromises had to be made in the hill-climbing algorithm in Chapters 3 and 4, which was unable to search the space of languages as thoroughly as desired (in particular, the undesirable constraint that mutant encoders had to be strictly more expressive than the champion).

As yet, there is no standard computational model for the evolution of language, which is not surprising given that the field is still trying to determine the relevant factors. Previous researchers have typically employed a model that is appropriate for demonstrating some phenomenon. The computational model used in this thesis is no exception: it represents another unique approach. However, there are some close similarities between the present model and those of Batali (1998) and Kirby (2000). In fact, Chapter 5 can be seen as a direct bridge between these two models.

In the simulations of that chapter, a simple recurrent network using the obverter procedure for production (Batali's learner) is placed in the context of a spatially arranged population with random death (Kirby's transmission dynamic). The positive results of Chapter 5 thus serve to highlight the generality of both Batali's and Kirby's results: Batali's learners can generate a learnable language within the Iterated Learning Model and Kirby's results are not critically reliant on a particular learning algorithm.

Despite the similarities in the learners of the present work and Batali's (1998) earlier work, the end results — in terms of the emergent languages — are quite different. Certainly there are significant differences between the transmission dynamic in Batali's populations and that here. However, the more likely determinant of this discrepancy in languages is the choice of semantic domain. Whereas Batali chose a combinatorial domain, the present work has used a continuous domain. In each case, the emergent languages are suited to the semantic domain. Such a difference is in accordance with Cangelosi's (2001) observation that an agent's interaction with the world plays a significant role in determining emergent linguistic structure.

In Kirby's (2000) simulations, populations started with no initial language. The language of a population was then bootstrapped via the process of random invention. In contrast, the starting languages in the systems considered in this thesis typically consisted of a very small number of unique utterances used to describe a wide variety of meanings (i.e., highly homonymous, poorly expressive languages). In Kirby's simulations expressive power came from finding common sub-terms between randomly invented utterances (structure was built in a 'top-down' manner). In the simulations of this thesis, expressive power typically came from extending or varying existing utterances for finer discriminations (structure was built in a 'bottom-up' manner). This difference between the two simulation frameworks is mirrored in the debates surrounding the evolution of human languages. The two competing theories in that field concern whether human proto-language was formed by the simplification of a set of complex, holophrastic utterances (where utterances are not compositional; Wray, 1998) or by combination of simpler elements (Bickerton, 1990). Interestingly, the simulation results presented here, combined with Kirby's results, suggest that both are viable alternatives.

The structures of the emergent languages from the simulations in this thesis generally had much in common with one another. Every language employed number-like structures. That is, there was a strong ordering of significance within an utterance,

based on the position of a symbol within the sequence. Each symbol then, came to have a meaning which was modified by its position within the sequence. This structure allowed learners to successfully generalise to symbols in novel positions. Learners needed only to learn the relative value of symbols (e.g., that `A` represented a value twice as large as `B`) and the relative value of positions (e.g., that the first position contributed one half of the meaning, the second position contributed one quarter of the meaning, etc.) to determine the meanings of novel utterances.

Kirby reasoned that the emergent structures in his simulations were a result of the bottleneck of linguistic transmission forcing languages to take on generalisable characteristics. This thesis proposed that for some learners, such as connectionist learners, there is an additional, *implicit* bottleneck. The implicit bottleneck is caused by constraints on learners' representations whereby it is more difficult to represent non-structured (random) languages. Thus, in the simulations presented in this thesis, languages were so structured partly because it was expedient for the agents. The presence of the implicit bottleneck can be observed by examining how the languages emerged. Languages started as highly homonymous and then gradually added structure to differentiate meanings. The highly homonymous languages at the start of the evolutionary process demonstrate that there was a strong similarity assumption (that similar meanings should have similar utterances) built in to the encoder; that there existed a bias towards treating similar meanings similarly.

The implicit bottleneck is effectively a functional constraint on language use and acquisition since it makes some languages more easy to represent than others. It may have been possible for networks to represent less structured languages, but it was less tractable for them to do so. The fact that languages became highly structured was thus in keeping with Kirby's earlier work (Kirby, 1999a) which demonstrated that weak functional constraints could, over time, dramatically reduce the range of observed languages.

6.4 Conclusions

This thesis has identified a variety of factors, beyond the constraints on the learner, that might act to constrain linguistic structure or which are important for enabling the emergence of linguistic structure from the dynamical system of language transmission.

- *The relationship between the biases of sender and receiver.* Linguistic structure

is constrained by the *intersection* of sender and receiver biases.

- *The communicative task.* The generalisation requirements of the learner (e.g., the different worlds of Chapter 4) alter the structural characteristics of the evolving language so that the language is best suited to the generalisation task.

- *The constancy of the learning environment.* The conditions in the simulations of Chapters 4 and 5 — whereby the learning environment was consistent for each learner — allowed languages to adapt to be learned from a particular experience.

- *The size of the 'bottleneck'.* While in Chapter 4 it was demonstrated that a language could be evolved to be learned from few examples, the simulations of Chapter 5 suggested that for the transmission dynamic to act as a *generator* of languages, the bottleneck had to be sufficiently large so that learners could reliably and consistently acquire the language.

- *The source of learners' linguistic data.* The simulations of Chapter 5 showed that the method for generating training data (i.e., the choice of speaker) could radically influence the likelihood of language emergence.

While many of the specific results presented in this thesis are independent of the connectionist paradigm, the use of connectionist models in the simulations permits some interesting observations that seem pertinent to connectionist natural language processing (CNLP). One of the criticisms of CNLP is the inability of connectionist models to generalise systematically (Fodor and Pylyshyn, 1988). In this thesis, languages were evolved to be learnable by a simple recurrent network. These evolved languages had quasi-systematic structures that are generalisable from few instances. In the type of framework used in this thesis (the evolutionary language perspective), learners need not be capable of acquiring arbitrary quasi-systematic languages. Instead, the language can adapt to the capabilities of the learner. Much work in the CNLP paradigm concerns probing the capabilities of connectionist learners. The results of this thesis suggest that the choice of learner is (to some degree) inconsequential; that language can adapt to fit the needs of the learner. In essence, the suggestion is that the question asked by CNLP — that as to what sort of learner is necessary to acquire human languages — needs to be considered in terms of how human language came to be in its present forms.

In the generative grammar framework, the range of human languages is directly constrained by UG. Thus, UG is a theory of both constraints on cross-linguistic variation and constraints on the language acquisition device. This thesis (and the evolutionary approach in general) has made a clear distinction between the two. The argument is that the constraints on the language acquisition device are filtered through the dynamics of language transmission. The constraints on cross-linguistic variation are therefore the end-point of a series of complex interactions, rather than a direct result of constraints on the learner. This thesis has examined the ways in which linguistic structure (and thus constraints on linguistic variation) can be shaped by those interactions.

The languages that have been produced by the agents during the course of simulations in this thesis have been far simpler than human languages. Such a result is unsurprising given the relative simplicity of both the simulated semantic domain and agents. Nevertheless, there are some interesting aspects of complex structure in the languages produced in the simulations that have some parallels with human languages. The first important observation is that the simulated languages *have* considerable internal structure, most easily demonstrated by the structural analyses of Chapters 3 and 4. In these structures, symbols could be interpreted as having specific meanings (typically either 'more' or 'less') which were dependent on their position ('a little more'; 'a lot more'). The structures of the evolved languages were quasi-systematic. Although there was a large-scale organisation to the languages there were many discrepancies. *These irregularities did not preclude learnability.* Thus, the evolved languages were highly structured, imperfectly regular and learnable by a general-purpose mechanism.

This thesis suggests that complex, semi-regular linguistic structures can adapt so as to facilitate its acquisition by general-purpose learning mechanisms. The implication for human language is that generative grammar's arguments for domain-specific, innate constraints on the human language acquisition device can be curtailed.

6.5 Further Work

The simulations performed in this thesis have, by necessity, used a highly abstracted, simplified, and idealised model of language, language user, and environment. These types of abstraction are fundamental to the modelling approach in that they allow the extrication of the relevant factors (and, of course, a tractable implementation).

The goal of the current research program is not (strictly) to describe the conditions of human language evolution, but rather to understand the general principles behind the emergence of language-like communication systems. The work presented in this thesis is therefore not directly aimed at capturing specific aspects of the human scenario, but at finding the critical aspects of the general situation. At present, no model of language emergence features all of the factors that are relevant to the emergence of language. Indeed, it seems likely that the field as a whole has not even identified all of the relevant factors.

This thesis has considered a variety of circumstances in which language emerges using a simplified model. As the complexity of the model increases, we should also expect an increase in the variety of factors that influence the emergence of language in the model. Thus, future work on the current model should consider increasing the complexity of learner, language and environment. Doing so should help enable exploration of other relevant factors in the emergence of language.

Bibliography

Bäck, T. and Schwefel, H. P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23.

Batali, J. (1994). Innate biases and critical periods: Combining evolution and learning in the acquisition of syntax. In Brooks, R. and Maes, P., editors, *Proceedings of the Fourth Artificial Life Workshop*, pages 160–171. MIT Press.

Batali, J. (1995). Small signaling systems can evolve in the absence of benefit to the information sender. Unpublished manuscript.

Batali, J. (1998). Computational simulations of the emergence of grammar. In Hurford, J. R., Knight, C., and Studdert-Kennedy, M., editors, *Approaches to the Evolution of Language*, pages 405–426. Cambridge University Press, Cambridge, England.

Batali, J. (in press). The negotiation and acquisition of recursive grammars as a result of competition among exemplars. In Briscoe, E. J., editor, *Linguistic Evolution Through Language Acquisition: Formal and Computational Models*. Cambridge University Press, Cambridge, UK.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Bickerton, D. (1983). Creole languages. *Scientific American*, 249(1):108–115.

Bickerton, D. (1990). *Language and Species*. University of Chicago Press.

Blair, A. D. and Pollack, J. B. (1997). Analysis of dynamical recognizers. *Neural Computation*, 9(5):1127–1142.

Blum, A. and Rivest, R. (1992). Training a 3-node neural network is NP-complete. *Neural Networks*, 5(1):117–128.

Bodén, M., Wiles, J., Tonkes, B., and Blair, A. D. (1999). Learning to predict a context-free language: Analysis of dynamics in recurrent hidden units. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, pages 359–364. IEE.

Briscoe, E. J. (1998). Language as a complex adaptive system: Coevolution of language and of the language acquisition device. In Coppen, P.-A., van Halteren, H., and Teunissen, L., editors, *Computation linguistics in the Netherlands 1997*, pages 3–40, Amsterdam. Rodopi.

Briscoe, E. J. (2000). Macro and micro models of linguistic evolution. Submitted to volume arising from the Third (Paris) Evolution of Language Conference.

Cangelosi, A. (2001). Evolution of communication and language using signals, symbols and words. *IEEE Transactions on Evolutionary Computation*, 5(2):93–101.

Cangelosi, A. and Parisi, D. (1998). The emergence of a language in an evolving population of neural networks. *Connection Science*, 10(2):83 – 98.

Casey, M. P. (1996). The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. *Neural Computation*, 8(6):1135–1178.

Chalmers, D. (1990). Why Fodor and Pylyshyn were wrong: The simplest refutation. In *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, pages 340–347.

Chalup, S. and Blair, A. D. (1999). Hill climbing in recurrent neural networks for learning the $a^n b^n c^n$ language. In Gedeon, T., Wong, P., Halgamuge, S., Kasabov, N., Nauck, D., and Fukushima, K., editors, *Proceedings of the Sixth International Conference on Neural Information Processing (ICONIP99)*, pages 508–513.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.

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

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press.

Chomsky, N. (1981). *Lectures on Government and Binding.* Foris, Dordrecht.

Chomsky, N. (1986). *Knowledge of Language.* Praeger, New York.

Christiansen, M. H. (1995). Language as an organism — implications for the evolution and acquisition of language. Unpublished manuscript.

Christiansen, M. H. and Chater, N. (1994). Generalization and connectionist languiage learning. *Mind and Language*, 9:273–287.

Christiansen, M. H. and Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, 23:157–205.

Christiansen, M. H., Chater, N., and Seidenberg, M. S. (1999). Connectionist models of human language processing: Progress and prospects. Special issue of *Cognitive Science*, 23(4):415-634.

Cleeremans, A., Servan-Schreiber, D., and McClelland, J. L. (1989). Finite state automata and simple recurrent networks. *Neural Computation*, 1:372–381.

Cohen, D. (1981). On holy wars and a plea for peace. *COMPUTER*, 14(10):48–54.

Darwin, C. (1890). *The Descent of Man and Selection in Relation to Sex.* John Murray, Albermarle Street, London, second edition.

Deacon, T. W. (1997). *The Symbolic Species: The Co-Evolution of Language and the Brain.* W. W. Norton and Company, New York.

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

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

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, 48:71–99.

Elman, J. L. (1995). Language as a dynamical system. In Port, R. F. and van Gelder, T. J., editors, *Mind as Motion: Explorations in the Dynamics of Cognition*, chapter 8. MIT Press.

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., and Plunkett, K. (1996). *Rethinking Innateness: A Connectionist Perspective on Development*. MIT Press, Boston.

Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, Special issue on Connections and Symbols:3–71.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58.

Gold, E. M. (1967). Language identification in the limit. *Information and Control*, 16:447–474.

Hadley, R. F. and Hayward, M. (1995). Strong semantic systematicity from unsupervised connectionist learning. In Moore, J. D. and Lehman, J. F., editors, *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, pages 358–363.

Hare, M. and Elman, J. L. (1995). Learning and morphological change. *Cognition*, 56:61–98.

Hashimoto, T. and Ikegami, T. (1995). Communication network of symbolic grammar systems. In Y. Aizawa et al, editor, *Proceedings of the International Conference on Dynamical Systems and Chaos*, volume 2, pages 595–598, Singapore. World Scientific.

Hashimoto, T. and Ikegami, T. (1996). Emergence of net-grammar in communicating agents. *BioSystems*, 38:1–14.

Hurford, J. R. (1989). Biological evolution of the Saussurean sign as a component of the language acquisistion device. *Lingua*, 77:187–222.

Hurford, J. R., Studdert-Kennedy, M., and Knight, C. (1998). *Approaches to the Evolution of Language: Social and Cognitive Bases*. Cambridge University Press, Cambridge, England. Volume arising from the first Evolution of Language conference.

Hutchins, E. and Hazlehurst, B. (1995). How to invent a lexicon: The development of shared symbols in interaction. In Gilbert, N. and Conte, R., editors, *Artificial Societies: The computer simulation of social life*. UCL Press.

Jackendoff, R. S. (1977). *X-bar syntax: A study of phrase structure.* MIT Press.

Jagota, A., Plate, T., Shastri, L., and Sun, R. (1999). Connectionist symbol processing: Dead or alive? *Neural Computing Surveys*, 2:1–40.

Kirby, S. (1998). Fitness and the selective adaptation of language. In Hurford, J. R., Knight, C., and Studdert-Kennedy, M., editors, *Approaches to the Evolution of Language*. Cambridge University Press, Cambridge, England.

Kirby, S. (1999a). *Function, Selection, and Innateness.* Oxford University Press.

Kirby, S. (2000). Syntax without natural selection: How compositionality emerges from vocabulary in a population of learners. In Knight, C., Hurford, J. R., and Studdert-Kennedy, M., editors, *The Evolutionary Emergence of Language: Social function and the origins of linguistic form*, pages 303–323. Cambridge University Press, Cambridge, England.

Kirby, S. (2001). Spontaneous evolution of linguistic structure – an iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, 5(2):102–110.

Kirby, S. (in press 1999b). Learning, bottlenecks and the evolution of recursive syntax. In Briscoe, E. J., editor, *Linguistic Evolution through Language Acquisition: Formal and Computational Models.* Cambridge University Press.

Knight, C., Hurford, J. R., and Studdert-Kennedy, M. (2000). *The Evolutionary Emergence of Language: Social Function and the Origins of Linguistic Form.* Cambridge University Press, Cambridge, England. Volume arising from the second Evolution of Language conference.

Kolen, J. F. and Pollack, J. B. (1990). Back-propagation is sensitive to initial conditions. *Complex Systems*, 4(3):269–280.

Lewis, D. K. (1969). *Convention: A Philosophical Study.* Harvard University Press.

Maass, W. and Orponen, P. (1998). On the effect of analog noise in discrete-time analog computations. *Neural Computation*, 10:1071–1095.

Maass, W. and Sontag, E. D. (1999). Analog neural nets with gaussian or other common noise distributions cannot recognize arbitrary regular languages. *Neural Computation*, 10(5):771–782.

MacLennan, B. J. and Burghardt, G. M. (1994). Synthetic ecology and the evolution of cooperative communication. *Adaptive Behavior*, 2(2):161–188.

McClelland, J. L. and Rumelhart, D. E. (1986). *Parallel distributed processing: Explorations in the Microstructure of Cognition, Vol 2: Psychological and biological models.* MIT Press, Cambridge, MA.

Moore, C. (1998). Dynamical recognizers: Real-time language recognition by analog computers. *Theoretical Computer Science*, 201(1–2):99–136.

Newmeyer, F. J. (1986). *Linguistic Theory in America.* Academic Press, second edition.

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science*, 14:11–28.

Niyogi, P. and Berwick, R. C. (1995a). A dynamical systems model for language change. Technical Report A.I. Memo No. 1515 and C.B.C.L. Paper No. 114, Massachusetts Institute of Technology Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Department of Brain and Cognitive Sciences.

Niyogi, P. and Berwick, R. C. (1995b). The logical problem of language change. Technical Report A.I. Memo No. 1516 and C.B.C.L. Paper No. 115, Massachusetts Institute of Technology Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Department of Brain and Cognitive Sciences.

Noble, W. and Davidson, I. (1996). *Human Evolution, Language and Mind: A Psychological and Archaeological Inquiry.* Cambridge University Press, Cambridge, England.

Nowak, M. A., Plotkin, J. B., and Jansen, V. A. A. (2000). The evolution of syntactic communication. *Nature*, 404:495–498.

Oliphant, M. (1999). The learning barrier: Moving from innate to learned systems of communication. *Adaptive Behavior*, 7(3-4):371–384.

Oliphant, M. and Batali, J. (1996). Learning and the emergence of coordinated communication. Submitted.

Pinker, S. (1994). *The Language Instinct.* William Morrow, New York.

Pinker, S. and Bloom, P. (1990). Natural language and natural selection. *Behavioral and Brain Sciences*, 13:707–784.

Pollack, J. B. (1987). *On connectionist models of natural language processing.* PhD thesis, Computer Science Department, University of Illinois, Urbana, IL.

Pollack, J. B. (1991). The induction of dynamical recognizers. *Machine Learning*, 7:227–252.

Rodriguez, P., Wiles, J., and Elman, J. L. (1999). A recurrent network that learns to count. *Connection Science*, 11(1):5–40.

Rohde, D. L. T. and Plaut, D. C. (1997). Simple recurrent networks and natural language: How important is starting small? In *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pages 656–661, Hillsdale, NJ. Erlbaum.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing: Explorations in the microstructure of cognition*, chapter 8, pages 318–361. MIT Press.

Rumelhart, D. E. and McClelland, J. L. (1986a). *On learning the past tenses of English verbs*, pages 216–271. In McClelland and Rumelhart (1986).

Rumelhart, D. E. and McClelland, J. L. (1986b). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1: Foundations.* MIT Press, Cambridge, MA.

Saunders, G. and Pollack, J. B. (1996). The evolution of communication schemes over continuous channels. In Maes, P., Mataric, M. J., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 580–589. MIT Press.

Savage-Rumbaugh, E. S. and Lewin, R. (1994). *Kanzi: The Ape at the Brink of the Human Mind.* John Wiley, New York.

Seidenberg, M. S. (1997). Language acquisition and use: Learning and applying probablistic constraints. *Science*, 275(5306):1599–1603.

Sejnowski, T. J. and Rosenberg, C. R. (1990). NETtalk: A parallel network that learns to read aloud. *Cognitive Science*, 14:179–211.

Senghas, A. (1995). *Children's contribution to the birth of Nicaraguan Sign Language.* PhD thesis, Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology.

Siegelmann, H. T. (1993). *Foundations of Recurrent Neural Networks.* PhD thesis, New Brunswick Rutgers, The State of New Jersey.

Steels, L. (1996). Emergent adaptive lexicons. In Maes, P., Mataric, M. J., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Sumulation of Adaptive Behaviour*, pages 562–567.

Steels, L. (1997a). The origins of syntax in visually grounded robotic agents. In Pollack, M., editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 1632–1641, Los Angeles. Morgan Kauffman.

Steels, L. (1997b). The synthetic modeling of language origins. *Evolution of Communication*, 1(1):1–34.

Sun, G. Z., Chen, H. H., Giles, C. L., Lee, Y. C., and Chen, D. (1990). Neural networks with external memory stack that learn context-free grammars from examples. In *Proceedings of the Conference on Information Science and Systems*, volume II, page 649, Princeton, NJ.

Tino, P., Horne, B. G., and Giles, C. L. (1995). Fixed points in two-neuron discrete time recurrent networks: Stability and bifurcation considerations. Technical Report UMIACS-TR-95-51 and CS-TR-3461, Institute for Advanced Computer Studies, University of Maryland.

Tonkes, B., Blair, A. D., and Wiles, J. (1998). Inductive bias in context-free language learning. In Downs, T., Frean, M., and Gallagher, M., editors, *Proceedings of the Ninth Australian Conference on Neural Networks*, pages 52–56.

Tonkes, B., Blair, A. D., and Wiles, J. (1999). A paradox of neural encoders and decoders, or, why don't we talk backwards? In McKay, B., Yao, X., Newton,

C. S., Kim, J. H., and Furuhashi, T., editors, *Simulated Evolution and Learning*, volume 1585 of *Lecture Notes in Artificial Intelligence*, pages 357–364. Springer.

Tonkes, B., Blair, A. D., and Wiles, J. (2000). Evolving learnable languages. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems 12*, pages 66–72. MIT Press.

Tonkes, B. and Wiles, J. (in press). Methodological issues in simulating the emergence of language. In Wray, A., editor, *The Transition to Language*. Oxford University Press.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

van Gelder, T. J. (1990). Compositionality: A connectionist variation on a classical theme. *Cognitive Science*, 14:355–384.

van Gelder, T. J. (1998). The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences*, 21:1–14.

Weckerly, J. and Elman, J. L. (1992). A PDP approach to processing center-embedded sentences. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, Hillsdale, NJ. Erlbaum.

Werner, G. M. and Dyer, M. G. (1991). Evolution of communication in artificial organisms. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artifical Life II*, pages 659–687. Addison-Wesley.

Wiles, J. and Elman, J. L. (1995). Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, pages 482–487, Cambridge, MA. MIT Press.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Wray, A. (1998). Protolanguage as a holistic system for social interaction. *Language and Communication*, 18:47–67.