PERGAMON

Neural Networks 12 (1999) 217–235

Neural Networks

Contributed article

Language acquisition from sparse input without error feedback

R.F. Hadley, V.C. Cardei

School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6

Received 27 June 1998; accepted 14 October 1998

Abstract

A connectionist-inspired, parallel processing network is presented which learns, on the basis of (relevantly) sparse input, to assign meaning interpretations to *novel* test sentences in both active and passive voice. Training and test sentences are generated from a simple recursive grammar, but once trained, the network successfully processes thousands of sentences containing deeply embedded clauses. All training is unsupervised with regard to error feedback – only Hebbian and self-organizing forms of training are employed. In addition, the active–passive distinction is acquired without any supervised provision of cues or flags (in the output layer) that indicate whether the input sentence is in active or passive sentence. In more detail: (1) The model learns on the basis of a corpus of about 1000 sentences while the set of potential test sentences contains over 100 million sentences. (2) The model generalizes its capacity to interpret active and passive sentences to substantially deeper levels of clausal embedding. (3) After training, the model satisfies criteria for strong syntactic and strong semantic *systematicity* that humans also satisfy. (4) Symbolic message passing occurs within the model's output layer. This symbolic aspect reflects certain prior *language acquistion* assumptions. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Connectionism; Language Acquisition; Active–Passive; Systematicity; Sparse input

1. Introduction

Children do not learn language in a vacuum. Frequently, a child encounters linguistic data in the context of situations that the child can, to some degree, perceive and conceptualize. The situational contexts in which linguistic utterances are encountered are widely thought to provide powerful semantic constraints on the learning process (see Pinker, 1984). Indeed, there is significant evidence that humans cannot learn even moderately simple context-free languages in the absence of accompanying semantic information (cf. Moeser and Bregman, 1973). For these reasons, researchers in language acquisition have often assumed that the learning agent (whether human or artificial) frequently manages to guess the speaker's intended meaning, thereby ensuring that some semantic information is available. Examples of systems which embody this assumption can be found in Anderson (1977), Pinker (1984), and St. John and McClelland (1990).

Now, although perceived semantic information certainly assists the learning process, acquisition of the active/passive voice distinction poses an interesting challenge. For, the distinction itself is not to be found in the external situations which a child perceives. The very same situation can be described by an active–voice sentence ('cats chase mice') or by a passive–voice sentence ('mice are chased by cats'). Moreover, the complexity of the learning process is compounded by the fact (stressed by Chomsky and others) that children learn language under conditions of *sparse* linguistic input. An important aspect of *sparse* input is that the set of sentences which a child encounters does not present all words in all syntactically legal positions. Indeed, it seems likely that most words the child encounters are *not* presented in all legal positions. (See Pinker, 1989, and Hadley, 1994a for 'field' examples of children understanding sentences containing words in positions which are *novel* to the child).

In Hadley (1994b) a definition of *strong semantic systematicity* is introduced. According to this definition, a cognitive agent or model exhibits strong semantic systematicity only if it can learn to assign appropriate meaning representations to sentences containing words in *novel* syntactic positions. In this context, a word is considered to occupy a novel position (e.g., grammatical subject) only if the agent has not encountered that word in that syntactic position at *any level of sentential embedding.* As shown in Hadley (1994a), humans *do in fact* satisfy this conception of semantic systematicity. Moreover, examples given in Pinker (1989) establish that humans display this property in the production and comprehension of both active and passive sentences.

Now, there do exist high-level symbolic algorithms for language acquisition which, at first blush, appear to satisfy the requirements of strong semantic systematicity, and

PII: S0893-6080(98)00139-7

otherwise cope with conditions of sparse input. However, these algorithms typically require that words in the input (or training) corpus be labelled or ''tagged'' in some way. For example, the algorithms presented in Wexler and Culicover (1980), Pinker (1984), and Berwick (1985) all require that lexical items be tagged with a syntactic role (e.g., as in Berwick) or with an argument position (relative to the verb, as in Pinker). This ''tagging approach'' is entirely reasonable provided we bear in mind that the algorithms in question are incomplete – they presuppose some prior learning on the agent's part. One goal of our present research is to provide indications as at how this ''syntactic tagging'' might occur. Thus, results presented here could be viewed as supplementing traditional symbolic approaches towards language acquisition (more on this below).

It is noteworthy that some (AI-inspired) symbolic models of language learning do not attempt to accommodate sparse input. For example, the methods described in (Berwick and Pilato, 1985) assume that the input corpus is ''positionally exhaustive'', i.e., during learning, virtually all words are presented in every legal syntactic position. (Bear in mind that if the word 'cat', say, has previously been presented as direct object only in a simple sentence, then a later presentation of 'cat' as direct object in a relative clause would *not*, on the present usage, constitute appearance in a novel syntactic position. Cf. Hadley, 1994a.) Although symbolic models which require exhaustive input of this kind may provide certain valuable insights, it seems clear that they cannot approximate the input–output behaviour of human learners. The same difficulty arises in most connectionist work on language learning, however. Indeed, as argued in Hadley (1994a), several prominent connectionist systems appear to employ, in the relevant sense, positionally exhaustive training corpora. These systems include those of: Elman (1990), St. John and McClelland (1990), Chalmers (1990), and Pollack (1990).

Since 1994, several connectionists have designed systems tailored to satisfy the criterion of strong *syntactic* systematicity. One of these, owing to Phillips (1994), is restricted in scope to simple sentences (containing no relative clauses). Another (Niklasson and van Gelder, 1994) does not address language learning, but confines itself to a very narrow range of logic formulae with at most one level of embedding. By contrast, Christiansen and Chater (1994) have extended the work of Elman (1993) on predicting grammatical categories. They employ a (relevantly) restricted training corpus containing sentences generated from a moderately complex, context-free grammar. Like Elman, they augment their network's memory during training by increasing the number of hidden-layer units. Christiansen and Chater acknowledge that their network exhibits ''strong generalization'' only in a few syntactic contexts, however. In addition, Hadley has argued Hadley (1994b) that their network does not succeed at its primary task (prediction) in the relevant sense.

Miikkulainen (1996) suggests that his recent model satisfies the criteria for strong systematicity. However, based on his remarks, it appears that his system can successfully process words in novel positions only when they are precise synonyms of other words which the network has previously encountered in those same positions. This restriction would certainly violate the spirit of the definition of strong systematicity given in Hadley (1994a,b). Apart from that, Miikkulainen himself notes that his network could not be viewed as a *learning* model, since his training regime presupposes crucial prior syntactic knowledge. For example, the network must *already know* where clausal boundaries occur in the input sentences. It must also know, prior to training, the internal, distributed representations of embedded clauses as distinct from main clauses.

One connectionist system, owing to Hadley and Hayward (1997), clearly *does* satisfy the criterion of strong semantic systematicity and learns from a sparse input set. In contrast with the networks cited above, the Hadley–Hayward model does not employ any form of error-feedback, but learns by purely Hebbian methods. During training, only one-third of the nouns are presented in all syntactic positions. Despite this, the network generalizes its capacity to interpret *all* nouns in novel positions, and generalizes to novel levels of clausal embedding. However, the Hadley–Hayward model can succeed only in the presence of active-voice sentences. The model which we offer below incorporates several techniques employed by Hadley and Hayward, but augments the network architecture in significant respects. These modifications enable the acquisition of a broader range of syntax. Notably, the new model learns to assign meaning representations to both active-voice and passive-voice sentences, *and can generalize this capacity to novel levels of embedding.*

Acquisition (or in some cases *processing*) of the active-passive distinction has previously been addressed by some connectionists (Chalmers, 1990; Miyata et al., 1993; St. John and McClelland, 1990). However, their work did not address the 'sparse input' requirement, and did not satisfy the criteria for strong semantic systematicity. Moreover, in the case of St. John and McClelland, the training of the connectionist networks involved explicit cues indicating whether each of the *training sentences* was in active or passive-voice[1]. As noted above, such indications are not present when a pre-school child encounters active and passive sentences in the course of language learning. Setting aside this aspect, though, I wish to stress that these connectionist systems were not intended to address sparse input or to satisfy strong systematicity.

In what follows, we present a ''connectionist-inspired'' system which both learns to assign meaning representations

[1] St. John and McClelland (1990) employ a special *bit* within their target *meaning* representations to indicate whether that meaning is expressed by an active or passive sentence. As they also employ an intensive form of error feedback during training (namely, backpropagation), this target bit value functions as an explicit cue, reflecting the voice of the input sentence. In all backpropagation trained networks, target output values function as one source of input to the training algorithm.

S → NP V-form NP

NP → N | N RC

N → women | girls | birds | bats | men | boys | chairs | balls | dogs | tables
| cats | mice

V → chase | sees | swing | love | avoid | follow | bump | hit | consume | dislike

V-form → V | are V by

REL-PRO → that

RC → REL-PRO V-form NP

Fig. 1. The Grammar of L.

to active and passive sentences *and* addresses the difficulties just mentioned. It should be noted, however, that our model employs certain classical techniques (such as ''message passing'' between nodes). It also assumes the existence of certain *innate language acquisition* mechanisms. Thus, in some crucial respects, our model conforms to the classical symbolic paradigm of language acquisition. The design of our model owes much, though, to prior *connectionist* work. Thus, our approach could be described as ''hybrid-connectionist'', but this might suggest to some readers that our system employs separate connectionist and classical modules. A better description, we think, would be to say that we employ *connectionist-inspired functionalism.* We use connectionist methods to implement high-level, abstract functions, but these functions were partly inspired by prior connectionist work on language acquisition and parsing.

Before proceeding to describe the details of our approach, it may be useful to summarize some of the goals achieved and results obtained. In particular,

- The model learns on the basis of sparse input. During testing, when clausal embedding can be as deep as level three, the number of possible sentences is well above 100 million. Yet, the network is successfully trained on a corpus of about one thousand sentences, where clausal embedding is restricted to at most level one. Thus, the training corpus is *far* from exhaustive.
- During training, two-thirds of all nouns are *not* presented in all legal positions. However, during testing, those nouns are each presented in positions novel to those words. The resulting novel sentences are each processed with complete accuracy. Over 16 thousand sentences have been tested.
- The model exhibits strong syntactic and strong semantic systematicity. Although sentences in the training corpus are generated from a simple grammar, the grammar is recursive. Following training, the system successfully processes substantially deeper levels of sentence embedding than occur during training (thus attaining level 4 in Niklasson and van Gelder (1994) generalization hierarchy).
- The network learns to assign meaning representations to both active and passive sentences (in simplified syntactic form). The active–passive distinction is acquired without external cues (or flags) that signal whether a given ''target meaning'' belongs to an active or passive sentence.
- Network learning involves both self-organizing (Grossberg, 1976) and Hebbian-inspired training[2]. It is widely believed that these are both closer to biological reality than the commonly used method of backpropagation of error. Also, in an important respect, the training employed is unsupervised. For the network is never provided with any form of 'error feedback'.
- Once training is complete, the network not only displays strong semantic systematicity, but a straightforward explanation of this fact exists.

As should be apparent from the points just cited, we have aimed for cognitive plausibility in some important respects. However, we emphasize that our model is not intended to be a full-fledged, complete model of language learning in humans. Our hope is that the approach presented here may provide certain *clues and techniques* which will be of value in the general study of language acquisition. As previously mentioned, our model might be used to augment those classical, symbolic, language-acquisition algorithms which require that input words be tagged with labels or ''argument positions''. Alternatively, the ''semantic parse'' structures which comprise the output of our fully trained network might serve as input for other connectionist language learning networks.

2. System task and goals

A multi-layer connectionist network (hereafter known simply as the 'network') is given the task of *learning* to produce ''semantic parses'' (or meaning representations) for both simple and complex sentences. The project consists of a training phase and a test phase. During each of these phases, both simple (main) clauses and embedded (relative) clauses may contain either active or passive verb forms.

Distinct training and test corpora are generated using the grammar of *L* (see Fig. 1). Note that in adopting this

[2] A variety of connectionist learning algorithms now exploit Hebb's basic proposal Hebb (1949) that the strength on a connection between two network units should be strengthened only when those two units are simultaneously active.

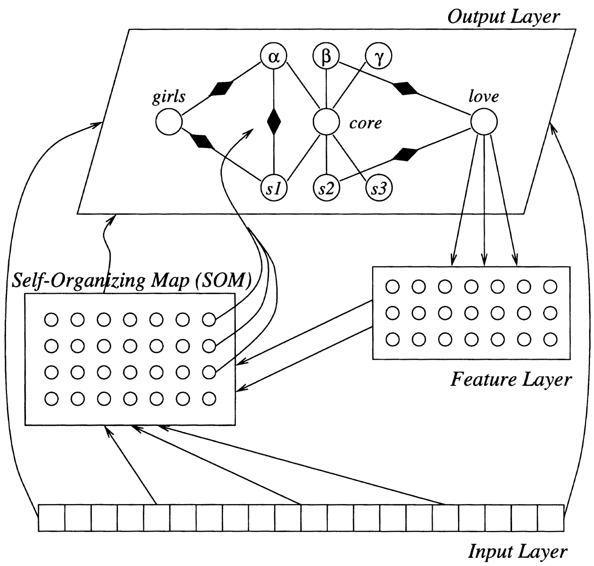

Fig. 2. Overview of Network Architecture. Arrows between layers correspond to entire sets of links. Within the output layer, there are concept nodes (e.g., 'girls' and 'love'), conceptual role nodes (α, β, γ) and sequence nodes ($s1$, $s2$ and $s3$). The blackened diamonds are nodes which serve to bind together pairs of attached nodes. There are many other links and nodes that occur within the semantic layer, but are not shown here. Further details will emerge in discussions which follow.

grammar, we follow the example of St. John and McClelland (1990) and Chalmers (1990) in that we employ a simplified syntax; we have not addressed the acquisition of past tense endings during passivization.

During the training phase, sentences are presented to an input layer one word at a time, and as training progresses, associations are learned between word (or lexical) nodes in the input layer and semantic nodes in the top-level, output layer. (see Fig. 2.) As will emerge in later sections, other associations must also be learned. In order for this learning to occur, we have assumed that, during training, the learning agent manages to *guess* the target meaning representations for the comparatively simple sentences in the training corpus.

Target (or ''guessed'') meaning representations include connectionist nodes representing both *concepts* and the *roles* which those concepts can play (such as agent or patient). These appear in the *output layer*, which also contains binding nodes and sequence position nodes. The latter enter into complex structures with concept and role nodes. Complete details are presented in the following sections, but for the present let us note that the learning agent's ''guessed representation'' requires only that the agent guess which concepts are bound (or attached) to certain thematic roles. The learner's ''guess'', *per se*, does not involve sequence position nodes. Eventually, sequence nodes are prompted to form bindings with other nodes. However, this occurs through unsupervised spreading activation, and makes no demands on the learner's capacity to guess meanings.

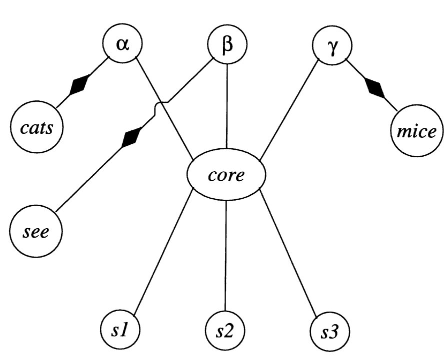

Fig. 3. A partial representation for the internal meaning of 'cats see mice'. Bindings to the nodes S1, S2 and S3 result later, from spreading activation.

Now, as noted in our introduction, the assumption that 'target meaning representations' are available to a learning agent figures prominently in several existing language acquisition models. Admittedly, the assumption is an idealization, but a good case can be made (Pinker, 1984) that children often, at least, manage to divine their parents' intended meaning.

3. Overview of network representations, architecture, and strategy

As previously mentioned, our system involves two major phases – a training phase and a test phase. Both phases assume the prior existence of representational structure within the output layer (see Fig. 2). Thus, we have assumed (as others have, including Pinker, 1984; McClelland and Kawamoto, 1986; St. John and McClelland, 1990), that the agent possesses at least a primitive ability to conceptualize external situations. In our model, this ability involves *concept* nodes (that can represent objects or actions) and *role* nodes (that can represent agent, action, or patient roles).

During the training phase, a sequence of sentences is presented, seriatum, to the network's input layer. As each sentence is presented (one word at a time), the target meaning representation for the *entire* sentence remains active within the output layer. This entire meaning representation is active throughout the processing of the given sentence. An example representation, for 'cats see mice', is shown in Fig. 3. Further details will emerge below, but for the moment, observe that nodes labelled 'cats', 'see', 'mice' are representations of the concepts involved, and nodes labelled α, β, and γ represent the roles 'agent', 'action', and 'patient', respectively. The dark, diamond-shaped nodes are 'binding nodes' (cf. Cottrell, 1985; Smolensky, 1990; Stevenson, 1994). Binding nodes are used to *bind* nodes together into a unified representation. Thus, the fact that 'see' is bound to β indicates that the 'see' concept plays the action role in the entire proposition being represented. The node labelled 'core' serves as a focal point for the entire proposition being represented, and binds together the various role nodes that belong to a single proposition.

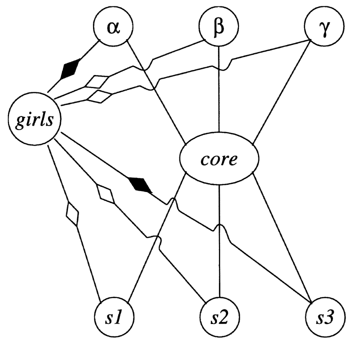

Fig. 4. The concept node, ‘girls, is shown in active bindings with the α and S3 nodes. Inactive, but potential bindings are also shown. Active bindings are indicated by blackened diamonds.

Nodes labelled ‘S1’, ‘S2’ and ‘S3’ are *sequence* nodes which eventually record the order in which other nodes become activated and bound. Note that in Fig. 3, the sequence nodes have not entered into bindings with other nodes. This is because, within the learner’s initial guess at the intended meaning, nothing is known about the sequence of activation that will emerge.

By contrast with the training phase, the *test* phase for a given sentence initially contains no active concept, role, or binding nodes. However, as successive words are presented to the input layer, activation propagates to the output layer, where the activation continues to spread, causing nodes to become active and bindings to occur. As various concept, role, and sequence nodes enter into bindings, *a semantic parse* results. This parsing process relies strongly upon proper training of four sets of links, viz., links between the input and output layers, between the input layer and SOM (the Self-Organizing Map), between the feature layer and SOM, and between SOM and the output layer (see Fig. 2). In addition, the parsing process relies upon certain principles of message passing and node-typing that, for present purposes, may be regarded as innately endowed. These principles are *not* specific to a particular language, and might well be intrinsic to the pre-existing conceptualizing ability that permits the learner to guess intended meanings during the training phase. These *message passing principles* are explained in Section 8.

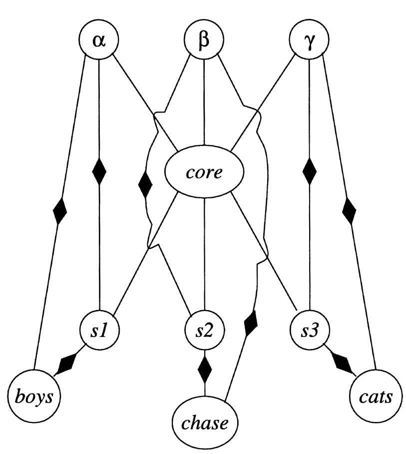

Fig. 5. A semantic parse structure for the sentence ‘boys chase cats’. Only active binding nodes are shown here.

3.1. Representations in the input layer

The input layer is a simple, one dimensional array of 25 units. Each unit represents a unique lexical item (word of vocabulary). As the words of a given sentence are fed to the input layer in sequence, the lexical unit corresponding to that word is activated and remains active just until it fires to the higher layers. Only one lexical unit is active at any given time. Further details are given in Section 4.

3.2. Representations in the output layer

In order to gain an understanding of the overall functional roles of various layers within our network, it is necessary to understand more fully the representations that reside within the output layer. Referring again to Fig. 3, we see that the core node has direct links to the three *conceptual role* nodes (α, β, and γ) and to the three sequence nodes (S1, S2, and S3). These links are bi-directional. The entire cluster, consisting of the core, links, three role nodes and three sequence nodes, is described as a ‘pnode’, which is our abbreviation for *propositional node constellation.* Between each role node and each sequence node, there exists a binding node, which may or may not be active. If active, we say that the given role (say, agent) is *bound* to a certain sequence position.

Within the output layer, there are a total of four pnode constellations. Usually, not all of these pnodes will be active when a sentence is assigned a ‘semantic parse’. However, during the test phase, when deeply embedded clauses may be present, it can happen that all four pnodes become active during the parsing of a single sentence. The function of a single pnode is to represent the meaning of a single, simple proposition, in which concepts are bound to each of the three conceptual roles (α, β, and γ).

As previously mentioned, the output layer also contains nodes that (locally) represent concepts of objects and actions. These concept nodes are activated, during the training and test phases, in response to activation flowing from the input layer. All concept nodes belong to a single, winner-take-all (WTA) network. For this reason, only one winner is selected to become ‘‘newly active’’ at a given time. Once activated however, *concept nodes can remain active* (subject to decay) *while other concept nodes are made active during later ‘‘parsing’’ of a given sentence*. Any given concept node can potentially enter into a binding with any given role node and any given sequence node. (See Fig. 4). Corresponding to each of these potential bindings is a binding node and links that connect the binding node to the pair of nodes that can potentially bind. An actual (or active) binding occurs only when the given binding node surpasses its ‘firing threshold’ (see Section 5.4 for details). Fig. 4.

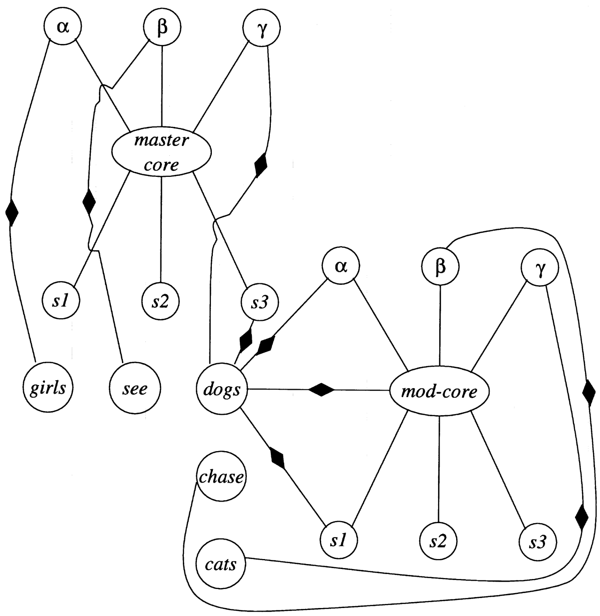

Fig. 6. A semantic parse structure for the sentence ‘girls see dogs that chase cats’. Only active binding nodes are shown here. For simplicity, bindings to most of the sequence nodes have been omitted from this figure.

Shows the concept node ‘girls’ in active bindings with the α (agent) node and S3 (sequence-3) node of a given pnode. Active binding nodes are indicated by blackened diamond-shaped nodes. The remaining diamond nodes are inactive, but represent potential bindings.

The function of sequence nodes at a given pnode is to enter into bindings with both concept nodes and role nodes in a particular sequence. We have assumed that all sequence nodes in a given pnode are hard-wired to become active in a particular sequence. (Note, however, that sequence nodes might also be *trained* to activate in a particular sequence. See Hadley and Hayward, 1995, for one such example.)

Fig. 5 shows all the active bindings required for a parse of ‘boys chase cats’. Note that, conceptually speaking, ‘boys’, ‘chase’, and ‘cats’ have the respective roles of agent, action and patient. Bindings between concept nodes and role nodes reflect this fact. Moreover, active bindings between the three concept nodes and the three sequence nodes reflect the sequential positions that those constituents occupy in the sentence ‘boys chase cats’. In addition, the active bindings between role nodes and sequence nodes reflect the fact that the constituent playing the agent role occupies first position in the sentence that has been parsed. Analogously, the action role node is bound to the sequence node for position two, and similarly for the patient node and S3. If the original sentence had been in passive voice, as in ‘cats are chased by boys’, the agent node, α, would be bound to S3 instead of S1, and γ would be bound to S1 instead of S3. Bindings between the concepts and sequence nodes would likewise be switched, but bindings between concepts and role nodes would be just as shown in Fig. 5.

Clearly, the task of setting bindings between role nodes and sequence nodes is crucial to correct functioning of the ‘test phase’. During testing, many of these bindings are set by activation that flows from the self-organizing map, while other bindings are set via spreading activation within the output layer. During training, all role-sequence bindings are set via spreading activation just within the output layer. This difference between training and test phases is caused by the differences in acquired thresholds (full details are given in Section 5.4).

Now, given that each pnode contains only three sequence nodes, it may appear *puzzling* that our fully trained network is able to parse deeply embedded clauses. However, as will emerge, entire pnode constellations can become attached (or bound) to a single concept node. This concept node, in turn, can be bound to a single sequence node in a given pnode. Thus, our post-training network is capable of constructing structures resembling deep trinary trees. The details of this process are presented in Section 8, but we may now examine how different pnodes can *represent* embedded propositions within a complex meaning structure.

Recall that each of the four pnodes that reside within the output layer can have bindings with various concept nodes, and thereby represent entire propositions. When relative clauses are present in a sentence, embedded propositions *modify* one or more concepts. Thus, in the sentence ‘girls see dogs that chase cats’, an embedded proposition ‘dogs chase cats’ modifies the concept ‘dogs’ that occurs in the main (top-level) proposition ‘girls see dogs’. Within our output layer, one pnode (termed the *master pnode*) is always involved in representing the main *proposition* that a sentence expresses. The three remaining pnodes (called *mod-pnodes*), can each, upon occasion, be involved in representing propositions that modify a concept in the main proposition. Mod-pnodes have a structure identical to the master pnode except that the core of each mod-pnode is linked to binding nodes that connect to the concept nodes. Thus, each mod-pnode core has a potential active binding with any given concept. Fig. 6 shows a mod-pnode actively bound to a concept node that occurs in another proposition. Note that this concept is bound to sequence nodes and role nodes in both the master pnode and the mod-pnode. (For simplicity of display, bindings involving most of the sequence nodes are not shown here, although they would exist).

In Fig. 6, both the main and embedded clauses are in active voice. If the embedded clause had been passive, as in ‘girls see dogs that are seen (see) by cats’, then the ‘dogs’ concept node would be bound to the patient node (γ) in the mod-pnode cluster, but would still (after parsing) be bound to S1 in that same cluster. However, as mentioned, during training, the learner’s ‘guess’ does not set any bindings between concepts and sequence nodes. Rather, those bindings are set indirectly by spreading activation. In addition, *training* sentences never involve more than one relative clause. So, at worst, the learner must sometimes guess that one simple proposition modifies a concept in another simple proposition.

Apart from pnode aspects that have already been described, one other detail should now be noted. The core of each pnode is linked to a unique satellite node, called the *focus of attention* node. These focus nodes can become active when a given core is prompted to fire by spreading activation within the pnode cluster. Thus, focus nodes can measure whether a given pnode is the currently the focus of recent activity. As explained in later sections, active focus nodes can 'gate' the flow of activation into a given pnode from the self-organizing map (SOM).

3.3. Representations in the feature layer

As indicated in the foregoing discussion, our approach assumes that the learner has already acquired concepts for certain objects and actions (or relations). These concepts are represented by local (atomic) nodes in the output layer, We assume, however, that in the processes of acquiring these concepts, the agent has tacitly discovered that certain features typically characterize the objects or relations being conceptualized. The functional role of the *feature layer* (Fig. 2) is to represent the agent's knowledge of features that characterize objects and relations. To this end, the feature layer contains 35 nodes, each of which represents a unique feature that is associated with a concept node in the output layer. For example, the node for 'cat' in the output layer has strongly weighted links to nodes representing the following features in the feature layer: [animate, four-legs, meows, has-weight, has-size, has-shape, has-location, furry, small, light, flat-face, flexible, small-nose]. In fact, each concept node in the output layer has links to every feature node in the feature layer. (A list of all features is given in Appendix A.) We assume that weights on these links were tuned during the process by which the learner acquired all the concepts represented locally in the output layer. For simplicity, we have pre-set each of these weights to a value of one or zero, to reflect whether the corresponding feature is strongly associated with the concept or not. Admittedly, this is an over-simplification (as indeed is our set of features). As frequently occurs in cognitive modeling, we have chosen to explore certain complexities in our model while simplifying others.

Now, each feature node in the feature layer has links to each node within SOM. In the early stages of network training, particular lexical nodes in the input layer rapidly become associated (i.e., weight-attuned) to particular concept nodes in the output layer. Once this association process stabilizes, a given input word will activate its proper conceptual correlate (node) in the output layer. This newly activated node, in turn, will send activation to the feature layer, thereby activating the set of features which are strongly correlated with the given concept node. Once this set of feature nodes becomes active, they fire and spread activation to SOM. This provides one source of learning for SOM.

3.4. Representations in SOM

SOM is a one dimensional array consisting of 40 nodes (anything in this range works well). It is fully connected, by trainable links, to both the input layer and the feature layer. During the training phase, both the input layer and the feature layer send activation to SOM, but not simultaneously. In an unsupervised fashion, SOM forms distinct neighborhoods that represent various activation patterns that SOM receives from the input array and from the semantic feature array. In consequence, SOM learns to reflect either the most recent *word* of input, or salient semantic properties of that word in the event that the word activates some concept in the output layer.

Details of the self-organizing (S-O) training are presented in Section 4. For now, we should note that input from the semantic feature layer causes SOM to develop two major representational neighborhoods, where each neighborhood has fuzzy boundaries. These two neighborhoods are distributed activation patterns that represent, respectively, the common features shared by object concepts and those shared by relational concepts (including actions). In more detail, concepts that share several semantic features (e.g., concepts of physical objects) will, upon separate occasions, activate overlapping sets of features in the *feature layer*. These overlapping sets will eventually, via S-O training, map into overlapping regions in SOM. The intersection of these overlapping regions, in turn, forms a neighbor-hood that comprises a distributed representation for the core properties of the semantically related set of concepts.

Of course, overlapping semantic regions can be found within the *feature* layer itself. As a consequence, one may wonder whether SOM is strictly necessary. For example, would it not be possible to create links directly from the feature map to the output layer, and simply remove both SOM and its incoming and outcoming links? The answer is ''perhaps, but not felicitously''. We must remember that SOM receives input both from the feature layer and from the input layer. Importantly, information received from the input layer is *not* relayed upwards if *semantic* information is soon received from the feature layer. The point of SOM is that it has a dual role – on the majority of occasions, activation conveyed upwards from SOM reflects semantic information *only* about core properties of the most recently activated concept node. On other occasions SOM conveys lexical information about the most recent input word. For this reason we view SOM as a lexical/semantic layer.

3.5. Links between SOM and the output layer

As noted earlier, within the output layer, there exists a unique binding node between each ordered pair of role nodes and sequence nodes in a given pnode cluster. These binding nodes (called 'bind-RS nodes' or simply 'RS nodes', for role-sequence binders) are grouped in separate winner-take-all (WTA) clusters. In particular, the three bind-RS nodes attached to any given role node form a WTA cluster. This reflects the fact that a role node cannot be actively bound to more than one sequence node at a time.

A major task of our network is to learn to activate the bind-RS nodes in a manner that reflects the correct role that a noun should play in a given sentence. For example, in the passive sentence, 'cats are chased by dogs', the concept for 'dogs' occupies the third sequence position among concepts that will be activated in a sequential reading of that sentence. Thus, since 'dogs' plays the agent role in this sentence, sequence node S3 should be bound to agent node α. Importantly, representations developed in SOM can assist in setting role-sequence bindings of this kind. Indeed, the current function of SOM in our post-training network is to set all such bind-RS nodes (though other functions may be discovered). For this reason, each node in SOM is connected to each of the bind-RS nodes. During the training phase, appropriate bind-RS nodes are activated via unsupervised activation flow. Simple Hebbian training between SOM representations and active RS nodes then permits the requisite associations to be learned for setting active-voice and passive-voice RS binding nodes.

It it noteworthy that, once training is complete, activation from SOM representations can cause several bindings to be revised simultaneously. Moreover, SOM representations can, in principle, reflect lexical information about word suffixes, including 'noun endings' (declension). Consequently, free word-order languages, which rely on noun declension to determine case roles, should be amenable to the approach adopted here.

3.6. Overall functionality

Now that the representational function of each layer has been sketched, the overall functionality of our model can be summarized as follows. (Please refer to Fig. 2).

1. Hebbian training of links between the input layer and the output layer ensures that strongly weighted associations are learned between referential words (the nouns and verbs) and corresponding concepts in the output layer. In addition, associations are discovered between the relative pronoun ('that') and the presence of mod-pnode cores.
2. As the words of a sentence are presented to the input layer, various concepts are activated within the output layer, and activation spreads both within that layer and to the feature layer. Spreading activation within the output layer ensures that some bindings occur. Activation reaching the feature layer causes appropriate feature nodes to fire into SOM.
3. Self-organizing training of links between the input layer and SOM enables SOM to develop representations that correspond to each of the vocabulary items. Similarly, S-O training between the feature layer and SOM creates distributed representations within SOM that roughly encode core properties of physical objects, physical actions, and some more abstract relations, such as 'sees' and 'loves'.
4. Hebbian training of links between SOM and RS nodes enables associations to be learned between certain self-organized representations and particular role-sequence binding patterns. The result is that very strong associations are discovered between those SOM representations that encode *passive-voice lexical indicators* (such as 'are' and 'by') and those role-sequence binding patterns that correspond to passive-voice semantic parses. Weaker associations are learned between SOM representtions of semantic feature clusters and those role-sequence binding patterns that correspond to active-voice semantic parses.
5. During the test phase, spreading activation from the input layer activates concept nodes in the output layer. These in turn spread activation in the output layer, thereby triggering appropriate concept-sequence bindings. Simultaneously, activation flows from concept nodes to the feature layer and thence to SOM. SOM then spreads activation to the output layer, and activates relevant RS nodes. Further spread of activation within the output layer results in appropriate concept-role bindings. The end result is that relevant concepts are bound to appropriate sequence nodes and appropriate role nodes.

4. The training corpus

Training and test data are generated using the grammar of *L* (see Fig. 1). The training corpus consists of about 1000 sentences. In recognition of the fact that active voice constructions are more common in ordinary dialogue than passive voice, we have restricted the occurrence of passive voice to roughly 24% of the training sentences. Within this 24%, passive constructions may occur either in the top-level or in relative clauses (i.e., either in main or embedded clauses).

In addition, less than 25% of the training sentences contain relative clauses. As a result of this, opportunities for training mod-pnode formations, which are central to our meaning representations for relative clauses, are less frequent than those for training the master pnode. Opportunities for training passive-voice settings of RS nodes within mod-pnodes are even less frequent.

Furthermore, and crucially, eight of the twelve nouns in language *L* are presented only in a single syntactic role during training. Four of these nouns can occur only as grammatical subject and four only as grammatical object. These eight nouns never appear, during training, as the head of a relative clause, since that would often entail that the noun plays one syntactic role within the main clause, and another role within the embedded clause. During the test phase, however, all these restrictions are dropped.

5. Network training

Successful network training requires less than eight complete passes through the corpus of approximately 1000 sentences. Each sentence of the training corpus is presented to the input layer one word at a time. When a given word is presented, its corresponding lexical unit is activated (set to $+1$) and remains active just until it fires. By contrast, the entire target meaning representation of the sentence being processed is active throughout the processing of a given input sentence. This means that all concept nodes and pnode elements involved in this meaning representation remain active during this time span. In particular, the relevant concept nodes and *pnode cores* are set to their maximum activation levels ($+3$). Role nodes are also active in this representation since the learner's 'guess' is that certain concepts are playing certain roles. These active role nodes are set to $+2$ initially, though they reach a higher activation level when prompted by spreading activation. In each complete propositional representation, the master pnode is active. If a relative clause is present in the input sentence, one of the three mod-pnodes will be chosen at random to represent the embedded proposition that corresponds to the meaning of the relative clause.

All links which enter concept nodes and pnode cores from the input layer are trainable links. Indeed, all links between layers are trainable, except the pre-weighted links that occur between the output layer and the feature layer. Links that occur between nodes *within* the output layer are not trainable, but serve to spread activation and foster bindings. The fact that not all links in our network are trainable reflects the fact that we have employed connectionism to implement higher-level functions. However, as previously noted, some of these functions are themselves inspired by well-known connectionist methods. The following subsection presents an example of this.

5.1. Training between the input and output layers

Training of links between the input and output layers employs the learning algorithm introduced in Hadley and Hayward (1995). This algorithm incorporates the basic Hebbian principle of incrementing weights on links that connect any pair of simultaneously active nodes. In the present case, simultaneous pairs of nodes occur each time a word-unit is activated in the lexical layer. For, each such unit is connected to every concept unit and every pnode core in the output layer. (Concept nodes and pnodes cores are collectively termed the 'semantic nodes'). When a given word-unit is activated, several semantic nodes are active in the output layer, namely, all those involved in representing the entire target proposition. Each link from the active word-unit to each active node in the output layer is incremented by a simple competitive-Hebbian formula (details in Appendix B).

A noteworthy point is that as these links are trained, their terminus (semantic) node acquires a firing threshold. At any given time, the terminus node is assigned a threshold equal to.8 of the largest input surge the node has ever received. As a result of this evolving threshold, after a few hundred input sentences have been processed, only those input nodes which are highly correlated with an active semantic node can cause that node to fire.

By the time training is complete, links connecting lexical input nodes to their correct, conceptual correlates within the output layer have all acquired weights which reflect the desired correlation. Indeed, weights on links connecting *referential word-units* (the nouns and verbs) to their proper conceptual counterparts are typically about 100 times stronger than weights that reflect spurious co-occurrence. In the case of 'that', which correlates strongly with mod-pnodes, the ratio of correct weights to spurious weights is not as dramatic, but it is more than sufficient to ensure that 'that' always (and only) excites modifier pnodes. (Recall that there are three mod-pnodes. As these are randomly chosen for inclusion in complex propositions, no single mod-pnode is invariably present when 'that' is present).

Significantly, lexical units which represent the words 'are' and 'by' do not have conceptual correlates within the output layer. For, it seems implausible that a learner would, prior to training, already possess concepts that correspond to the meanings of 'are' and 'by', which merely serve to indicate the passive voice in our simplified language, *L*. Indeed, the role of these passive indicators seems more syntactic than referential.

5.2. Training of SOM

The training of SOM is somewhat unusual in that links enter SOM from two separate layers, viz., the input layer and the feature layer. We employ here a version of S-O training owing to Kohonen (1984), and this requires that all links entering SOM be tuned simultaneously. Thus, each time activation spreads to SOM, whether from the input layer or from the feature layer, the entire set of weights, W, on links entering SOM is tuned. When an input node is activated, it spreads activation to both SOM and the output layer. At this precise time, nodes in the feature layer all have an activation of zero, because the feature layer is reset to zero each time a new word of input is about to be processed. However, activation spreading from the input layer to SOM triggers S-O learning, as described in Appendix B.

As this occurs, activation reaching the output layer triggers a WTA competition among just the concept nodes and pnode cores (i.e., 'the semantic nodes'). When a mod-pnode core wins this competition, activation spreads within the output layer, as detailed later in Section 5.3. As the mod-pnode cores are not atomic concepts, and lack the semantic

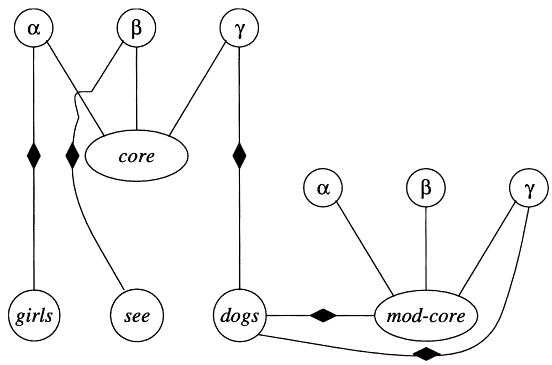

Fig. 7. The concept node, 'dogs', is shown in bindings with role nodes within a master pnode and a mod-pnode. This much would be part of the learner's ''guess''.

features of concepts, these core nodes are not connected to the feature layer. Thus, the firing of a mod-pnode core does not spread activation to the feature layer and does not indirectly trigger S-O training. However, when a *concept* node wins the WTA competition, it fires, thereby spreading activation both inside the output layer and to the feature layer. Activation reaching the feature layer causes some semantically relevant subset of feature nodes to become active and to fire along links to SOM. This in turn triggers S-O training on the entire set of weights, W, that enter SOM. At this point, there are active nodes in the feature map, but the input layer is no longer active. Thus, on separate occasions, SOM has an opportunity to develop representations corresponding to the semantic features of object concepts and action concepts.

5.3. Activity within the output layer

Soon, we shall want to consider details of training on links between SOM and active RS nodes in the output layer. Before doing so, we need to understand how and why particular RS nodes become active during training. This, in turn, requires an understanding of spreading activation and binding processes within the output layer. These processes are triggered as follows:

As each word is presented to the input layer, activation is propagated upwards to semantic nodes in the output layer. Each semantic node receives activation equal to the weight on the link coming into that node from the active input node. Upon receiving this input surge (or 'boost'), all concept nodes and mod-pnode cores enter into a WTA competition. (The master pnode core does not compete. It will be active in every proposition and does not play a semantic role in the usual sense). That semantic node whose received boost is largest will win the competition. We have assumed that semantic nodes will spend their excess 'boost' when competing, but not their initial level of activation. As the maximum *stable* activation level for semantic nodes is $+3$, any residual boost value is not retained once competition is complete.

Now, the winning semantic node will be either a concept node or a mod-core. When either of these wins, it fires, but there are different ramifications in each case. We shall consider each of these cases. However, first observe that some activation will have reached the master pnode core from the input layer. This activation causes the master core to fire. This, in turn, activates the first sequence node, S1, in the master pnode cluster.

5.3.1. When a concept node wins

When the winner of the WTA is a *concept* node, it fires, thereby spreading activation to the feature layer. Concurrently, the firing of the concept node spreads activation within the output layer. Recall that during training, all concept nodes involved in the target proposition are already active and are in active bindings with relevant role nodes. These role nodes are also initially active (at $+2$ activation).

Now, even in early stages of training, the winning concept node will be among the initially active concept nodes. (This is primarily as a result of the fact that no concept node ever has the opportunity to accumulate *more* weight, on the relevant link, than the correct (appropriate) concept node, and the latter is always active in the target meaning representation). Also, given that a winning concept node *fires*, it spreads activation towards binding nodes. As a consequence, the winning concept binds with the only available sequence node, S1. Moreover, activation will spread to the binding node that already binds the concept to an active role node. (See Fig. 3.) This active binding node is thereby prompted to fire into the attached role node, which, as a consequence, jumps from its current activation level ($+2$) to a higher level ($+3$). We now have a highly active role node and a highly active sequence node, S1. Each of these will spread activation towards binding nodes. The result is that they will bind, since they are the most active (and currently available) role and sequence nodes within the master pnode cluster. The bind-RS node involved in this new binding attains an activation level of $+2$. (Recall that role nodes can only be bound to one concept node and one sequence node at a given time — similarly for sequence nodes. Also, note that during both training and test phases, activation levels decay over time. More on this below).

Once the active sequence node, S1, has entered into bindings with both a concept node and a role node, it is ''fully satisfied'' and can enter into no other bindings unless some existing binding is broken. (As things naturally occur, no bindings are broken during the training phase). In general, any new fully-satisfied sequence node will fire into the pnode core, prompting the core to activate the next sequence node, if any remain. Thus, once S1 is fully satisfied, it prompts the core to activate the S2 sequence node.

Now, in certain cases (as in Fig. 7), a winning concept node (e.g., 'dog') has active bindings to role nodes in both a master pnode and a mod-pnode. In such cases, when the winning node fires, it not only spreads activation within

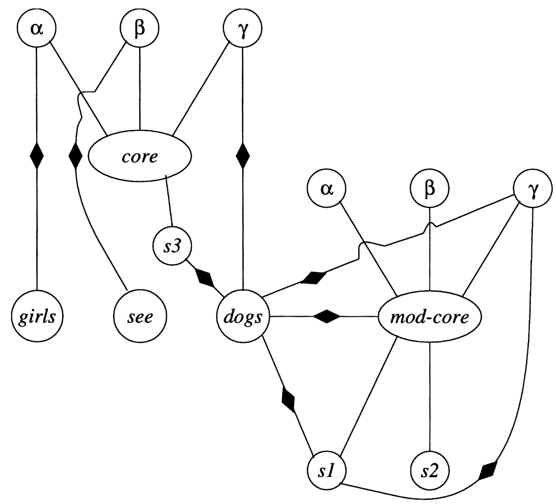

Fig. 8. Here we see the result of bindings that occur after ‘dogs’ has fired and spreading activation has triggered further bindings.

the master pnode, as described above, but it spreads activation directly to both the role node and mod-core that are contained within the mod-pnode cluster. (See Fig. 7). When activation reaches the mod-core, it prompts the core to fire and activate the first sequence node, S1, within the mod-pnode cluster. Once S1 becomes active, it will seek to bind with the most active available concept node and the most active role node within its own cluster. For this reason, ‘dogs’ would now bind with S1 in the mod-pnode cluster. In addition, the recently ‘‘boosted’’ role node would now bind with the most active sequence node, S1, just as described above. Finally, once the S1 node in the mod-pnode cluster is fully satisfied, it would prompt the mod-core to activate a new sequence node, S2. Thus, the total net effect of the firing of ‘dogs’ would be that ‘dogs’ binds with a sequence node in the master pnode and also within the mod-pnode. In addition, spreading activation causes role nodes to bind with sequence nodes within both pnode clusters, and S2 becomes active in the mod-pnode cluster, as shown in Fig. 8.

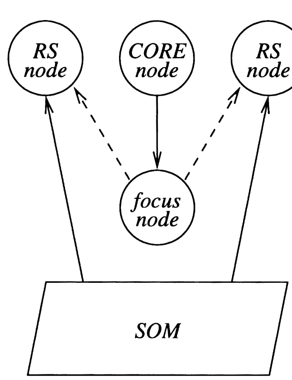

Fig. 9. A focus node is attached to each core node. Each of the virtual links (the dotted lines) leaving a focus node makes a conjunctive connection at the point that each link from SOM enters a RS node (i.e., any RS node that exists within the pnode cluster that ‘‘owns’’ the focus node). Only when activation flows along both of the conjoining links that enter a RS node can the RS node actually receive activation from SOM and only then does weight tuning occur on links from SOM to the RS node.

5.3.2. When a mod-pnode core wins

Having examined the case where a concept node wins the WTA competition, let us now consider what occurs when a mod-core wins instead. Except in the earliest stages of training, a mod-core wins a WTA competition because a lexical input node, corresponding to the relative pronoun ‘that’, has just spread substantial activation to a mod-core. This is a consequence of the fact that, early in network training, correct associations between lexical nodes and semantic nodes are learned.

Suppose for the moment, then, that a mod-core, which is an active component of the complete target semantic representation, has just ‘won’ and subsequently fired. Cores of pnodes have ‘‘outgoing links’’ only into their own sequence nodes and into their respective ‘‘focus of attention’’ nodes. As our given mod-core has presumably just fired, it spreads activation to its ‘‘focus node’’ and will also attempt to activate a sequence node, *unless* it has already activated a sequence node that remains, as yet, not fully satisfied. Let us assume that the mod-core fired as a response to a competition triggered by activation from the ‘that’ lexical node. This means that the preceding word was a noun, which should have caused a concept node to fire on the preceding iteration. The firing of this concept node would have reached the mod-core in the manner described in the preceding subsection. Thus, the mod-core would already have activated both its S1 sequence node (which would now be fully satisfied) and its S2 node, which remains unsatisfied.

5.3.3. The role of ‘‘focus nodes’’

In the foregoing, we have examined how various bindings are set within the output layer during training. Especially germane to our present concerns are the RS node settings. The setting of RS nodes determines whether, say, sequence position S1 is bound to an agent role or a patient role. During the test (or parsing) phase, RS nodes are set by activation flowing from active regions within SOM. Indeed, this also occurs in the latter stages of training, when RS node thresholds are surpassed by the sum of activation flowing from SOM. (This latter aspect will be explored in the following section). Our present concern is to stress that, during both training and testing, activation usually flows from SOM into *appropriate, active RS nodes*. In particular, flow of activation and weight training should occur on those links entering active RS nodes which belong to the pnode which is currently the center of activity *in the sense* that the core node is being stimulated by firing concepts or other satellites within the pnode cluster (such as role or sequence nodes). For, activity of this kind indicates that, cognitively speaking, the proposition represented by that ‘‘most active’’ pnode is the current ‘‘focus of attention’’.

The focus of attention nodes, mentioned earlier, serve the role of measuring which pnode within a complex propositional representation is currently receiving the most attention. This occurs as follows: Each time a core node receives stimulus from within its own cluster or, in the case of mod-cores, by winning a WTA competition, the core fires into its own unique ''focus of attention'' node (See Fig. 9). This focus node receives and assumes an activation level equal to that of the core node which just fired. In addition, all focus nodes, which are collectively attached to the various core nodes, belong to a single WTA network. As a consequence, the most active focus node at any given time will win a competition with all other focus nodes in the competitive cluster. The winning *focus* node retains the activation level it possessed at the onset of this competition, while the losing focus nodes are effectively inhibited by the winner. It sometimes happens that two focus nodes tie in this competition, in which case they both function as winners.

Now, winning focus nodes are able to *gate* activation flow, from SOM, into all active RS nodes that exist within the same pnode cluster as the winning focus node. Active focus nodes remain able to do this as long as they continue to win competitions. The actual *gating process* is implemented algorithmically in our current network because the connectionists methods by which such gating can be achieved are already well understood. However, the dotted lines in Fig. 9 indicate possible links which could achieve this gating process. These links may be regarded as ''virtual'' in our current implementation.

Now, it should be noted that the activation level of core nodes, which supply activation to focus nodes, decays over time. Thus, focus nodes attached to *newly* activated mod-cores will receive higher activation levels than focus nodes attached to previously activated core nodes. In addition, let us note that any active focus node becomes *inactive* if it receives an inhibitory signal from its attached core node. This always and only occurs when the core node receives a ''termination signal'' from the *last sequence node* within the given pnode cluster. That is, when the last sequence node (S3) in a pnode cluster becomes *fully satisfied*, it can enter into no further bindings. In this case, it sends a signal to the core, which then inhibits its own focus node. A focus node which is thus inhibited drops to zero activation and ceases to inhibit other focus nodes attached to other pnode cores. The latter focus nodes are then, once again, able to assume the activation levels of their attached core nodes, which triggers a new WTA competition among the focus nodes. In this way, another, less active pnode becomes the focus of attention[3].

[3] Clearly, the design and behaviour of focus nodes, in relation to core nodes and RS nodes, was inspired by a higher level, algorithmic process. We regard this algorithmic process as part of a high level, possibly innate, learning system. While we think it seriously implausible that a cognitive agent would possess *neural wiring* resembling the structures just described, it does not seem wild to suppose that a cognitive agent would incorporate the *functionality* of the algorithm we have implemented.

5.4. Training between SOM and the RS nodes

All nodes within SOM are fully connected to each RS node in the output layer. The initial weight on each of these links is 0.001, which is appropriate given that Hebbian training is employed. During training, weights are incremented on all of these links, provided (a) that the sending node (in SOM) fires, (b) that the relevant focus node ''gate'' is open (see Fig. 9), and (c) that the receiving RS node was already active, or now becomes active because its current *threshold* was surpassed by the sum of activation arriving at the RS node. Let us consider conditions (a) and (c) in turn; (b) was discussed in the preceding subsection.

(a). As previously mentioned, SOM receives activation both from the input layer, and, on most occasions, from the feature layer. The quantity and distribution of this received activation always determines a winning node within SOM. As each word of input is processed, SOM waits to see if activation will arrive from the feature layer. If it does arrive, a new winner emerges in SOM. Otherwise, the winner selected by the latest surge from the input layer is retained as the *current winner*. In either case, the current winner plays a special role, in that its activation level determines which SOM nodes may fire on the current iteration. In particular, any SOM node whose received activation is at least half of the winner's can fire upwards to the output layer. This firing triggers Hebbian training on links between SOM and the RS nodes.

(c). A ''receiving'' RS node may already be active because it recently effected a binding (between a role and a sequence node) as a result of spreading activation within the output layer. In such cases, the RS node will have its maximum activation level (+2) or a slightly lower level, as a result of decay. In addition, a RS node can become active if its current activation threshold is surpassed by arriving input activation. Thresholds of RS nodes vary over time, as a function of how much training has occurred on links entering the RS node. At any given time, a RS node's threshold equals (2 minus the sum of all weights entering the node). As each RS node has a maximum weight limit of +2 (i.e., the sum of weights entering the node can never exceed +2), a RS node's threshold begins near +2 and, over time, diminishes towards zero.

It is important to note that a RS node may be only *very briefly* activated because, though it received a sum of activation from SOM that exceeded the RS node's threshold, it then entered a WTA competition with other RS nodes, one of which had a higher activation level. Any RS node that was activated by spreading activation from a role and a sequence node (inside the output layer) would have a high activation level and would inevitably win such a competition. RS nodes which lose such competitions are not involved in the weight tuning process. Technical details of this Hebbian process are given in Appendix B.

5.5. Training phase – a second pass

In Sections 5.1–5.4, we examined training processes that occur in response to the processing of a single word of input. (Appendix B presents a high-level, algorithmic summary of these training processes.) In accord with conditions described above, this training occurs in response to each word in an input sentence. However, before another input sentence is processed, an additional training pass is made through the original sentence, while bindings in the output layer are held constant. That is, after the original sentence has been processed, all bindings in the output layer are frozen, and another training pass occurs. On this second pass, all weight increments are increased by a factor of five. Thus, the second pass is really equivalent to five further passes through the given sentence using normal increments. For reasons of efficiency, we have compressed these five passes into a single pass. Apart from increased efficiency, the motivation for our second pass is that it occurs when *all* relevant bindings are in place, rather than just a subset of relevant bindings.

6. The test corpus

The test corpus was intended to ensure that each of the eight nouns that were restricted to a single syntactic position during training would be tested in all other legal positions. To this end, a test corpus of 17 000 sentences was created. Of these, more than 16 000 contain relative clauses, both in active and passive voice. Around 3000 of the test sentences contain relative clauses at a deep level of embedding (level 3), and even more contain clauses at the second level of embedding. Active and passive voice formations are equally prevalent at all levels of embedding.

7. Results of testing

Our basic network was separately trained on five distinct training corpora, where each (1000 sentence) corpus conformed to the specifications described in Section 4. As a result, five distinctly weighted networks were obtained. Each of these five networks was then tested on the test corpus described immediately above. Sentences in the test corpus were presented, *seriatum*, to each of the five networks and the output representations were examined for correctness by a separate (classically symbolic) program. A given output representation was judged to be correct if (1) all appropriate nodes possessed activation levels greater than their firing thresholds, (2) all appropriate bindings were activated, (3) no inappropriate nodes were active. Our testing of all five distinct networks yielded 100% accuracy in each case.

8. The test (or parsing) phase

As mentioned earlier, some of the binding processes that occur during the test phase of our network are determined solely by activation levels that result from spreading activation. This spread of activation is determined by acquired weights (during training) and by the pre-existing link structure between layers and within the output layer. In addition, however, binding processes in the output layer are also influenced by certain general principles that are assumed to govern the ‘‘message passing’’ behaviour of the parsing processes that occur within the test phase. That is, in effect, the output layer is partially governed by a pre-existing set of parsing principles, which are not specific to any given language and which operate in parallel. Whether this pre-existing set of principles should be regarded as innate or as the result of prior learning can be left an open question for present purposes.

In order to understand our ‘‘minimal parser’’ and how it interacts with spreading activation levels to form coherent representations, we need first to understand that sequence, role, binding and concept nodes in the output layer each have a unique identity label, called an ‘‘id-tag’’. In our implementation, each id-tag is a unique numeral, where a portion of the numeral determines its ‘type’. However, for expository purposes, and in terms of functionality, it is easiest to imagine an id-tag as consisting of two parts: the node *type* and a numeral. For example, concept nodes have the symbolic type ‘c’, whereas role nodes have the symbolic type ‘r1’, ‘r2’, etc. The three role nodes at a given pnode, then, might have the following id-tags: r2-1, r2-2, r2-3. Role nodes at a different pnode might all have the symbolic type, ‘r3’. The situation is analogous for sequence nodes and also for binding nodes. Thus, sequence nodes at different pnodes have different symbolic types, but within a given pnode, they all have the same symbolic type. All concept nodes have the same symbolic type, ‘c’, but distinct numeral components. Note that when the type and numeral for each node is concatenated (as in s1-3), the result is a unique id-tag for the node.

Now that id-tags have been introduced, we can state certain general principles that govern the parsing process during testing. We present these principles with annotation, so that their motivation can emerge as appropriate. The principles are:

P1. Whenever two nodes bind, they exchange id-tags (via the connecting link) and retain a copy of the id-tag just received in a list of such tags. This applies to binding nodes as well: The result is that each role, sequence, and concept node ‘‘knows’’ the identity of all nodes they are bound to, including binding nodes.

P2. When two nodes bind, they not only exchange their own id-tags, but they exchange the list of id-tags that they currently have stored. Such lists are always short because binding processes result in ‘‘triangles’’ in which a concept, a role, and a sequence node are each bound to each other.

P3. (A derived principle). As a consequence of P1 and P2, binding nodes that create new bindings between two other nodes will often ''inherit'' a list of id-tags. This list may contain the id-tag of another binding node that has already entered into a binding with one of the involved nodes. In such cases, we call the earlier binding the ''ancestor'' binding. A newly activated binding node which inherits the id-tag of an earlier binding is called a ''descendant'' binding.

P4. A node cannot be in an active binding with more than one other node *of a given type*. Recall, however, that role (or sequence) nodes in different pnode clusters have different types.

P5. A node does not bind, even indirectly, with two distinct nodes of the same type. Specifically, if a node (N1) ''holds'' within its list of id-tags the tag of a node (N2) of a given type, T, then N1 is bound, either directly or mediately, to N2. In such cases, N1 will *not* bind with another node, N3, which has within its list of id-tags the tag of some node of that same *type* T (e.g., the role type), but which possesses a different identity number from N2.

P6. Role and sequence nodes can only bind with each other if they belong to the same pnode. However, concepts can enter into bindings in more than one pnode.

P7. Other things being equal, nodes prefer to bind with the *most active* node of a given type, provided this is permitted by principles P4, P5 and P6.

P8. When activation spreads from SOM to RS nodes, new WTA competitions arise. This can cause a formerly active binding node to become inactive. When this occurs, that binding node spreads its id-tag, along with a ''delete'' signal to the nodes it formerly joined. They will then delete that binding node's id-tag from their current list, and continue to propagate the ''delete-id-tag'' message so that other nodes may do the same.

P9. Whenever a descendant binding node, D, receives a ''delete id-tag'' signal from a binding node which D recognizes as an ancestor binding (caused by the presence of the id-tag in the descendant's current list), then D ceases to be an active binding. The motivation for this is that, as things occur, the descendant binding only arises as a side-effect of the ancestor binding, and conceptually, should cease to exist when the ancestor binding is broken.

P10. In general, whenever a binding is broken between any two nodes, they each delete the other's id-tag from their current list of tags.

It should be noted that when a binding is broken, the ''terminal nodes'' which were formerly bound are prompted to fire, and will thus seek new bindings, where possible. In general, a binding can occur only if (a) a binding node receives sufficient activation from a firing node, or (b) the binding node is already active. In the former case (a), the node which fires into the binding node may be within the output layer or within SOM. In cases when firing from SOM causes a RS node to win a WTA competition, that RS node can effect a binding between a role node and a sequence node, provided they are both already active. If only one of the pair is active, the RS node will bind with that active node. If neither is active, the binding node becomes inactive, since its role (like all binding nodes) is to effect bindings with active nodes.

With principles 1-10 in hand, we are now in a position to consider the *general cycle* of activity that occurs as each word of a test sentence is presented during the parsing (or test) phase. In presenting this cycle, we assume what our testing has already confirmed, viz., that inter-layer training does produce the weight vectors we anticipated.

In the outline presented below, binding processes occur at various points. Though we do not explicitly say this at every relevant point, each of these bindings involves the id-tag swapping processes, and binding preference principles described above.

8.1. Outline of the basic test cycle

1. The first word of the sentence activates a node in the input layer, which in turn sends activation to the output layer. Some of this activation reaches the master pnode core which now becomes highly active (+3). (Recall that the master core does not compete with other semantic nodes). Activation from the master core causes the first sequence node to become active (+3). In addition, the master core fires into its attached ''focus of interest'' node, which now assumes the activation level of the core.
2. Repeat while there are words left in the sentence

2.1 If the input word was a noun or verb, then we would expect the following to happen:

Activation reaching the output layer is sufficient to cause one or more concept nodes to surpass its (their) firing threshold. When this occurs, a WTA competition ensues among all such concept nodes, and one winner emerges.

That winning concept node now fires into the feature layer, with the result that some feature nodes become active. These in turn fire into SOM, where some neighborhood of nodes will become active.

Also, that same winning concept node now spreads activation within the semantic layer. This spreading activation reaches binding nodes. At this point, there is already at least one active, available sequence node connected to one of those binding nodes. A WTA competition occurs among all viable binding nodes, and a winner is chosen. This binding node remains active, effectively binding the most active concept node (our recent winner) to the most active sequence node that is free to bind. Note that id-tags are exchanged here and preference principles govern the process.

At this point, the active neighborhood within SOM can fire, thereby sending activation to all RS nodes that are currently ''enabled'' by an active focus of attention node.

WTA competitions now occur among each set of RS nodes that belong to each role site within the ''currently enabled'' pnode. Winning RS nodes are picked for each of these competitive sets.

Each winning RS node will retain the activation level which it has just received from SOM, *unless it already has a higher level of activation.* In general, active RS nodes always retain the higher of these two levels.

Each of these winning RS nodes is connected to a sequence node and a role node. If neither of these is already active, the RS node ceases to be active. However, if either the sequence or role node is already active, the RS node will bind with it. Usually, this means the RS node binds with an active *sequence* node. If such a binding occurs, then the RS node fires and sends activation towards any inactive role node to which it is connected. Normally, some role node now receives this activation, in which case the hitherto inactive (role) node now assumes its highest activation level (+3). Id-tag and preference processes apply here.

The newly active role node will now spread activation towards binding nodes that connect with concept nodes. Usually, this triggers a binding between the newly active role node and the most active concept node that is not already bound to another role node within the current pnode. (i.e., the role node binds with the most active *available* concept node.) Again, id-tag and preference processes apply here.

2.2. If the input word was 'that', then, within the output layer, we would expect the winner of the competition among semantic nodes to be a mod-pnode core. In this case, the mod-core will become highly active (+3) and fire, thereby activating its first sequence node (S1) and activating its focus of attention node. This *focus* node then enters a WTA competition with any other active focus node and would win (since any other focus node would have a lower activation, as a result of decay). Following that:

The mod-core binds with the most active concept node. The newly activated sequence node (within the mod-pnode) will now bind with the most active, *available* concept node.

The mod-pnode core has no connection to the feature layer, and so cannot fire into it. However, SOM waits for input from the feature layer. As none arrives, SOM will fire into all RS nodes that belong to the current mod-pnode. (This pnode is the current focus of interest, and so its RS nodes are currently able to receive activation.) WTA competitions now occur within each enabled cluster of RS nodes, and winners emerge, as previously described. Winning RS nodes will be selected, but since there is only one active sequence node (S1) within the mod-pnode cluster, only one of the winning RS nodes can now bind with a sequence node. When this binding occurs, the RS node fires and thereby activates a role node. Most often, this role node will be an α node, but it will sometimes be a γ node. The newly active role node will now bind with the most active, available concept node (which is the same concept node that just bound to a sequence node). At this point, no more bindings are possible, and the first sequence node is fully satisfied.

2.3. If the input word is not a noun or verb and not a relative pronoun ('that'), then, given our restricted vocabulary, it must be a passive voice indicator (either 'are' or 'by'). In this case we would expect that no semantic node wins a WTA competition because no semantic node will exceed its firing threshold. As a consequence, the feature layer will receive no input from the semantic output layer and will not fire into SOM. However, SOM still briefly waits for input from the feature layer. Once it is clear that such input will not arrive, SOM fires into the RS binding nodes. The result is that WTA competitions occur among all RS nodes that are currently enabled by an active 'focus of interest node'. If any RS node which now wins was *not already active*, then some previously active RS nodes may now become inactive. When this occurs, bindings are broken, and this will often trigger the breaking of descendant bindings, as previously explained. In addition, all newly activated (winner) RS nodes will fire, spreading activation towards role nodes and sequence nodes. If such activation reaches an active node, a new binding will occur. This in turn will often cause an attached role node to become active, as a result of the firing of the RS node when a binding occurs.

When no further bindings can occur, a decay factor is applied to all active concept, role, core, focus, and sequence nodes.

Also, if a sequence node became fully satisfied during the current cycle, then a new sequence node is activated at the current pnode (provided there remains a sequence node to activate).

If the *last* sequence node at a given pnode became fully satisfied during the current cycle, then the 'focus of interest' node at that pnode will be de-activated. At this point, the cycle ends, and a new input word is processed. Thus, we return to step 2 above.

9. Discussion

We trust that, by now, it is clear that the approach adopted here relies strongly upon techniques drawn both from the classical symbolic paradigm and from the connectionist school. The connectionist learning methods employed here play a crucial role in our model's success. Equally important are the prior link structures inherent in the output layer and the general principles governing id-tag exchange and binding possibilities. These pre-existing structures and principles are consistent with the general tenets of the Chomskian stance on language acquisition, viz., that language learning is possible only when prior linguistic universals and language acquisition mechanisms are present. However, as far as we have discovered, nothing in our approach precludes the possibility that these aspects could be acquired by prior learning.

In future work, we plan to address certain limitations of the present implementation. For example, our existing preference rules (P1-P10) collectively entail that a given concept node can be bound only to one role node within a given pnode. (Of course, the concept node may have bindings to roles in more than one pnode.) We believe this limitation could be removed through an improved set of preference rules. Also, our current implementation limits the maximum depth of clausal embedding to level three. This limit could easily be extended to level five by the addition of extra mod-pnodes and further training. However, this modification would not meet the concerns of those who maintain that a language acquisition system should have the capacity for unbounded recursion. To meet such concerns about *competence* (as opposed to matching *performance*), one could introduce a type-token distinction for mod-pnodes, and generate new copies of trained mod-pnodes as they are needed. This approach would also have the advantage of allowing just a single mod-pnode to be employed during training, with the result that training would be considerably more rapid. Given that we are not presently emulating *pure connectionism*, such a type-token strategy should present no obstacles. Further details on this approach are given in Hadley and Hayward (1997).

It is noteworthy that a type-token distinction among mod-pnodes could bring another significant benefit. For example, it would almost certainly mean that a significantly smaller fraction of passive voice sentences and relative clauses could be employed during training. As things now stand, our percentages for passives and relative clauses (24% and 25%, respectively) are too high for cognitive fidelity. Although we have not methodically investigated how much lower these percentages could be set before our model would fail, the fact that three separate mod-pnodes require training probably requires that these percentages be set above 10%. The difficulty at present is that, when only 1/4 of all relative clauses are in passive voice, and only 1/4 of training sentences contain relative clauses, then just 1/16 of training sentences present the passive-relative combination. When three separate mod-pnodes must be trained, there are comparatively few opportunities for training an individual mod-pnode in the passive-relative RS node configuration. As an aside, it is interesting to note that both our current model and the modification just outlined would predict that children make considerably more errors in the comprehension and production of passives within relative clauses than within main clauses. For, in either case, mod-pnodes have fewer opportunities for training in passive voice than the master pnode.

Apart from the foregoing, we anticipate other modifications which would expand the range of languages learnable by the our model. For example, we have performed related experiments, involving multiple context layers that record prior states of a different SOM, with encouraging results. The inclusion of such context layers in our current model could ensure that activation of RS nodes will reflect a larger preceding context (within the input stream) than is currently the case. This would render the setting of RS bindings more context-sensitive, and extend the range of learnable languages. In addition, we anticipate that the inclusion of prepositional phrases within the target language could be accommodated through the introduction of other kinds of pnodes. Note that prepositional phrases modify concepts in a fashion not unlike relative clauses.

Moreover, it should be noted that even the present implementation could learn a variety of word orders. Nothing in the present model creates a bias towards a NP-V-NP (noun-phrase, verb, noun-phrase) ordering. NP-NP-V and V-NP-NP orders are equally learnable. Also, as previously mentioned, it seems feasible that free word-order languages are learnable. Provided the grammatical role of nouns, within the input stream, is indicated by case markers (such as noun suffixes), SOM could detect these markers and correlate them with appropriate role-sequence bindings in the output layer.

In passing, we should address a claim made in our introduction, viz., that our model not only attains strong systematicity but does so in an intelligible manner. We believe that brief reflection reveals *how* the network attains systematicity. During the testing phase, at relevant steps, the most active concept node will bind with the most active role node. Moreover, the ability of a noun or verb to activate a new concept node in no way depends upon that word's syntactic context. All that matters is the strength of connections between the given input node and concept nodes in the output layer. By contrast (again during testing), the activation of role nodes is sensitive to the preceding sentential context, but not to the identity of particular nouns and verbs. Rather, role nodes are indirectly activated by SOM representations which *reflect very general semantic aspects* of nouns and verbs. Thus, role nodes can become active when syntactically appropriate. Now, because a concept node becomes active independently of surrounding context, its ability to bind with an active role node is not affected by restrictions on the syntactic positions during training, of any *word* which expresses that concept. The ability of a concept node to bind with a sequence node is analogously unaffected by the training phase. Consequently, the validity of semantic parses created during testing is unaffected by our restrictions on noun position during training. The upshot is strong systematicity.

10. Summary of results

We have presented a connectionist-inspired, parallel processing model for the acquisition of the active-passive distinction. Although the system has clear limitations, and does not constitute a cognitively accurate model, we suggest that the model's contribution consists partially in the techniques introduced, and the general direction of research outlined. Apart from that, the system's significance stems from its simultaneous satisfaction of several goals which, arguably, would need to be attained in a cognitively faithful model of the acquisition of the active-passive distinction. We know of no existing language learning system that satisfies even a majority of these goals.

These goals are:

- Meaning representations are accurately assigned to both active and passive sentences (albeit in a simplified syntax). The active–passive distinction is acquired without supervised provision of cues or flags (in the output layer) that indicate whether the internal ''target meaning'' belongs to an active or passive sentence.
- Relative to a vast space of possible test sentences, the model learns on the basis of sparse input.
- The model generalizes its capacity to interpret active and passive sentences to deeper levels of clausal embedding.
- After training, the model satisfies criteria for strong syntactic and strong semantic systematicity that humans also satisfy (cf. Hadley, 1994b).
- Training of the model employs only error-unsupervised learning methods. It is widely believed by psycholinguists that humans receive very little (if any) negative feedback during pre-school language acquisition.

In passing, we should note that our connectionist-inspired model is not intended either as a confirmation or as a counter-example to Fodor and Pylyshyn (1988) views on systematicity and the limits of connectionist representations. We freely acknowledge that much of our network is implementing higher levels of functionality. Indeed, this has been a major focus of this article. We also believe, though, that connectionist training methods contribute in essential ways to the model's success.

Acknowledgements

Robert Hadley gratefully acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (Grant No. A0899) during the period of this research. In addition, we are deeply grateful to the late Timothy Edgar for his implementational work and comments on a prototype of the model described here. We are indebted also to anonymous reviewers, enlisted by this journal, for their many helpful comments. Finally, we would like to thank Dirk Arnold for his work on the graphic display of figures.

Appendix A

Features assigned to concepts in our implementation are admittedly incomplete and approximate. However, they serve to convey the general approach we have adopted.

Features of concepts of physical objects are taken to be subsets of the following: animate, inanimate, two-legs, four-legs, talks, barks, meows, squeaks, has-weight, has-size, has-shape, has-location, furry, large, small, heavy, light, laughs, bites, long-snout, flat-face, small-nose, rigid, flexible, tubular, round.

Note that concepts of all physical objects would have certain of these features: e.g., has-weight, has-size, has-shape, has-location.

Features of concepts of actions and relationships (expressed by verbs) are taken to be subsets of the following: physical-motion, involves-contact, involves-animate, rapid, slow, emotive, feeling-nice, feeling-bad, involves-perceiving.

Appendix B

B.1. High-level algorithm for the training phase

Repeat for each sentence in the training corpus;

Activate those concept nodes, role nodes and binding nodes within the output layer which are required to represent the meaning of the given sentence;

Repeat for each word in the given sentence;

Reset all nodes in the input layer, feature layer, and SOM to zero activation.

Activate the lexical input node corresponding to that word;

Spread activation from the given input node to the output layer and to SOM.

Apply Hebbian training (item B.2, below) to links between the input layer and the output layer.

Apply S-O training (item B.3, below) to all links entering SOM.

If a semantic node within the output layer received activation which surpassed its threshold, them choose some winning semantic node within the output layer, allow it to fire, and perform bindings as described in Section 5.3.

If a winning node was selected in the previous step, it spreads activation to the feature layer. Activation then spreads to SOM.

Find the most active node within SOM and apply S-O training (item B.3, below) to all links entering SOM.

The most active node within SOM also defines an active neighborhood. This active neighborhood fires, spreading activation to those RS nodes within the output layer which are currently enabled by a focus of interest node.

Apply Hebbian training (item B.4, below) to links between SOM and enabled RS nodes.

End Inner Cycle.

Without altering any bindings within the output layer, repeat the above inner cycle once again for each word in the given sentence, using five-fold increments (see Section 5.5).

End Outer Cycle.

B.2. Hebbian training of links between the input and output layers

Let S be any active concept or pnode core (i.e., semantic node) in the output layer. Then the link (L) coming into S (from an active input node) is incremented as follows:

$$\text{increment} = 0.0005R$$

where R is the ratio of the current weight on L to the current average weight of links into S. Note that the ratio R can cause links with 'above average' weight to be rewarded significantly. As learning proceeds, the learning on winning links accelerates, and little weight is assigned to links which reflect spurious co-occurrences.

Initially, the weight on each link, L, is 0.001[4]. All semantic nodes, N, in the output layer have a fixed weight limit (of $+1$) on the sum of weights on links entering N. As a result of this, weight modification between the lexical layer and the output layer must halt when all semantic nodes have reached their weight maximums, i.e., when no concept node or pnode core has any weight left to distribute among its incoming links. This occurs well before other network training has completed. Once the lexical-to-semantic links have been fully trained, they continue to spread activation to the output layer as other network training progresses.

B.3. Self-organizing training of the SOM: technical details

We have obtained good results with both one-dimensional and two-dimensional SOMs of various sizes. For simplicity, our present network now employs a one-dimensional map containing 40 nodes. Weights on links entering SOM (the set W) are set to random values between zero and 0.01. Nodes within SOM are initialized to have equal values; these sum to $+1$.

As activation flows into SOM, some node within the map receives the greatest activation. This node (the 'winner') is used as an origin point for measuring Euclidean distances. Weights in the set W are then updated according to the following function:

$$\Delta w(k,j) = \eta e^{-ra}(I(j) - w(k,j))$$

In the above, $w(k,j)$ is the weight on the link from input node j to node k; η is the learning rate (1.0 in our case); r is the Euclidean distance between node k and the winner node; a is the annealing factor; and $I(j)$ is the value of input node j. In our application, the annealing factor, a, has an initial value of 0.005 and increases linearly during training towards 1.0. This causes eventual stabilization of SOM, as a result of the fact that each winner's neighborhood diminishes in size as training progresses. By the time a few thousand sentences are processed (about 2 epochs), the neighborhoods are effectively stable and frozen.

[4] Random weights close to 0.001 would serve equally well.

B.4. Hebbian training of links between SOM and RS nodes

Weight modification is applied to every link connecting a node within SOM (designated as node i) to a RS node (denoted j) when conditions (a), (b) and (c) obtain. The precise increment applied to each link is determined by the Hebbian rule given below. This weight modification occurs only after activity in the output layer has stabilized (i.e., WTA competitions are complete and bindings in the output layer have all been set).

$$\Delta(i,j) = \eta R \text{ activation}(i) \text{ activation}(j)$$

where $\Delta(i,j)$ represents the size of the increment; R is a user determined learning rate (0.00005 in our implementation); and η is an adaptive rate that increases over time once neighborhoods in SOM have largely stabilized. The η rate begins near zero and increases gradually during the first training epoch (around 1000 sentences). During each succeeding epoch, η increases in increments of roughly $+1$ until it reaches a maximum of at most seven.

References

Anderson, J. R. (1977). Induction of augmented transition networks. *Cognitive Science*, *1*, 125–157.

Berwick, R. C. (1985). The acquisition of syntactic knowledge. Cambridge, MA: MIT Press.

Berwick, R. C., Pilato, S. F. (1985). Reversible automata and induction of the English auxiliary system, Proceedings of the Ninth International Joint conference on Artificial Intelligence, Los Angeles, 880-882.

Chalmers, D. (1990). Why Fodor and Pylyshyn were wrong: the simplest refutation. Proceedings of the Twelfth Annual Conference of the Cognitive Science Society, Cambridge, MA. 340-347.

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

Cottrell, G. (1985). Connectionist parsing. Proceedings of the Seventh Annual Conference of the Cognitive Science Society, Irvine, CA. 201-211.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–212.

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

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

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: Part I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, *23*, 121–134.

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

Hadley, R. F. (1994b). Systematicity revisited: reply to Christiansen and Chater and Niklasson and van Gelder. *Mind and Language*, *9*, 431–444.

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

Hadley, R. F., Hayward, M. B. (1995). Strong semantic systematicity from unsupervised connectionist learning. Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society, Pittsburgh, PA. 358–363.

Hebb, D. O. (1949). The organization of behaviour. New York: Wiley.

Kohonen, T. (1984). Self-Organization and associative memory. Heidelberg: Springer Verlag.

McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: assigning roles to constituents of sentences. In D. E. Rumelhart & J. L. McClelland & PDP Research Group (Eds.), *Parallel Distributed Processing; Explorations in the Microstructure of Cognition*, 2. Cambridge, MA: MIT Press.

Miikkulainen, R. (1996). Subsymbolic case-role analysis of sentences with embedded clauses. *Cognitive Science*, *20*, 47–74.

Miyata, Y., Smolensky, P., Legendre, G. (1993). Distributed representation and parallel processing of recursive structures. Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society, Boulder, Colorado, 759–764.

Moeser, S. D., & Bregman, A. S. (1973). Imagery and language acquisition. *Journal of Verbal Learning and Verbal Behavior*, *12*, 91–98.

Niklasson, L. F., & van Gelder, T. (1994). On being systematically connectionist. *Mind and Language*, *9*, 288–302.

Phillips, S. (1994). Strong systematicity within connectionism: the tensor-recurrent network. Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, Atlanta, GA, 723–727.

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

Pinker, S. (1989). Learnability and cognition: the acquisition of argument structure. Cambridge, MA: MIT Press.

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, *46*, 77–105.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*, 159–216.

Stevenson, S. (1994). A unified model of preference and recovery mechanisms in human parsing. Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, Atlanta, GA, 824–829.

St John, M. F., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, *46*, 217–257.

Wexler, K., & Culicover, P. (1980). Formal principles of language acquisition. Cambridge, MA: MIT Press.