The role of recurrent networks in neural architectures of grounded cognition: learning of control

Frank van der Velde[1], Marc de Kamps[2]

[1] Leiden Institute for Brain and Cognition, Leiden University, 2333 AK, Leiden, The Netherlands (corresponding author)
e-mail: vdvelde@fsw.leidenuniv.nl
[2] Biosystems Group, School of Computing, University of Leeds, LS29JT Leeds, United Kingdom
e-mail: dekamps@comp.leeds.ac.uk.

Abstract. Recurrent networks have been used as neural models of language processing, with mixed results. Here, we discuss the role of recurrent networks in a neural architecture of grounded cognition. In particular, we discuss how the control of binding in this architecture can be learned. We trained a simple recurrent network (SRN) and a feedforward network (FFN) for this task. The results show that information from the architecture is needed as input for these networks to learn control of binding. Thus, both control systems are recurrent. We found that the recurrent system consisting of the architecture and an SRN or an FFN as a 'core' can learn basic (but recursive) sentence structures. Problems with control of binding arise when the system with the SRN is tested on number of new sentence structures. In contrast, control of binding for these structures succeeds with the FFN. Yet, for some structures with (unlimited) embeddings, difficulties arise due to dynamical binding conflicts in the architecture itself. In closing, we discuss potential future developments of the architecture presented here.

Keywords: grounded representations, binding control, combinatorial structures, neural architecture, recurrent network, learning

1. Introduction

An example of a recurrent network used as a neural model of language processing is presented in figure 1. It consists of a 'Simple Recurrent Network' (SRN), used by Elman (1991) in a word prediction task. An SRN is a three-layer network, in which the activation of the hidden layer is fed back to the input layer. Typically, the activation in the hidden layer produced by a given word in a sentence is then used as a (partial) input with the next word in the sentence. In the word prediction task, the SRN is trained on a set of sentences, to predict the next word that would appear given a sentence context. In this way, an SRN can learn to predict that in the context *boys who dog sees* a plural verb (e.g., *chase*) will follow.

Dagstuhl Seminar Proceedings 08041
Recurrent Neural Networks- Models, Capacities, and Applications
http://drops.dagstuhl.de/opus/volltexte/2008/1421

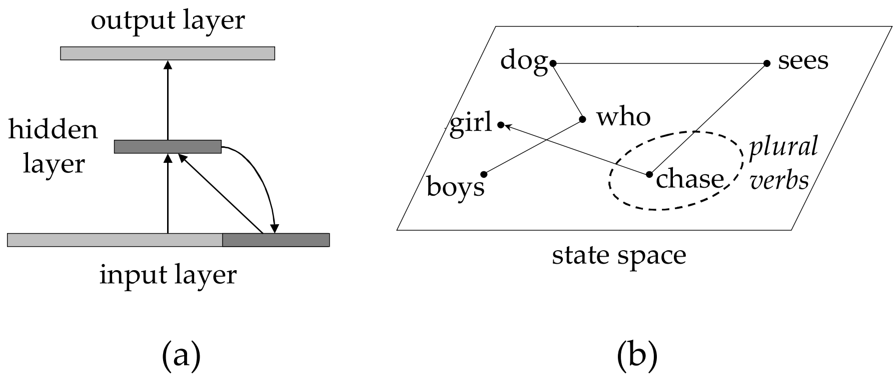

Fig. 1. A simple Recurrent Network (SRN) predicts the next word type in a sentence like *boys who the dog sees chase the girl.* (a) Structure of the SRN. (b) the state space of the connections in the network (between the hidden layer and the output layer) when the sentence is processed.

However, despite this initial success, language processing with SRNs is faced with serious limitations. In particular, SRNs cannot handle the (combinatorial) productivity of natural language (e.g., van der Velde, van der Kleij van der Voort, & de Kamps, 2004). Furthermore, a prediction of the next word type in a sentence is not really what language processing is about. Instead, the purpose of language is (at least) to provide 'who does what to whom' information. For example, for the sentence *boys who dog sees chase girl* a question like "Who do the boys chase?" should be answered by a specific answer (*girl*), not by an indication of a word type (*noun*).

Furthermore, a major aspect of language processing in the brain is the nature of (word) representation. We argue that representations of words/concepts in the brain are grounded in a network structure, related to aspects of word meaning derived from perception, action, emotion and semantic information. Because representations are grounded, they cannot be copied and pasted to form combinatorial structures like sentences (as in symbolic architectures). To ensure that representations remain grounded, they cannot be encoded (encrypted) either (as in the state space of an SRN, figure 1b), to form combinatorial structures. Recently, we presented a neural architecture that produces sentence structures based on grounded word representations (van der Velde & de Kamps, 2006). The architecture can handle the (combinatorial) productivity of language, and it answers 'who does what to whom' questions in a specific way. The architecture consists of neural 'binding' mechanisms that produce (novel) sentence structures on the fly. Here, we discuss the role of recurrent networks in this architecture. In particular, we discuss how the control of the binding process can be learned using recurrent network structures. First, however, we briefly discuss grounded representations and the architecture for sentence structure.

2. Grounded representations

Figure 2a illustrates the grounded representation for the word *cat*. It consists of a network that interconnects all aspects related to the word (concept) *cat*. This includes all perceptual information related to cats, action processes related to cats (e.g., the experience of stroking a cat, or the ability to pronounce the word *cat*), all emotional content associated with cats, and all other information related to or associated with cats, such as the semantic information that a cat is a pet, or the (negative) association between cats and dogs.

The grey oval in the center, labeled *cat*, plays an important role, because it interconnects the neural structures in the representation of *cat*, and because it can be connected to sentence structures, as in *The cat is on the mat* (thus, it can embed the word structure of *cat* in sentence structures). However, it would be wrong to see this oval itself as the neural representation of *cat*, because its representational value derives entirely from the network structure of which it is a part. Therefore, even if it were technically possible to make a copy of the oval (e.g., of its internal network structure or pattern of activation) and transport that to another location, it is useless to do so. When the internal network structure or pattern of activation of the oval is copied and transported, it is separated from the remaining network structure of the representation *cat*, and thus it loses its representational value. For example, if this copied and detached oval were connected to the structure of a sentence, it would not represent the word *cat* in that sentence (or any other word).

The grounded nature of word representation in the brain has been investigated in a number of studies. For example, Vigliocco et al. (2006) found modality-specific brain activation during word comprehension, in which motor related words (nouns and verbs) activated motor related brain areas and sensory related words (nouns and verbs) activated sensory related areas. Tettamanti et al. (2005) observed an activation difference between action related words (verbs), in which parts of the premotor cortex that code for specific actions (related to mouth, hand or leg actions) were also activated by the words describing these actions (e.g., *bite*, *grasp* or *kick*). Because Tettamanti et al. presented the words in sentence contexts (e.g., *I bite an apple*, *I grasp a knife* or *I kick the ball*), the observed word activations strongly suggest that the representations of the words remain grounded when the words are part of sentences (word combinations).

2.1. Conditional connections

The labeled connections in figure 2 are of a special kind. In a connection between two neurons, activation flows from the pre-synaptic neuron to the post-synaptic neuron, when the pre-synaptic neuron is active. This kind of connection is associative, because activation flows without any form of control. In contrast, the labeled connections are conditional: activation flows only when the condition indicated by the label is met. For example, the activation of the label is, e.g., by the query “What is a cat?”, opens the connection between the network structures for *cat* and *pet*, so that *pet* can be given as an answer to this query (van der Velde & de Kamps, 2006). Conditional connections are needed to represent relations in network structures

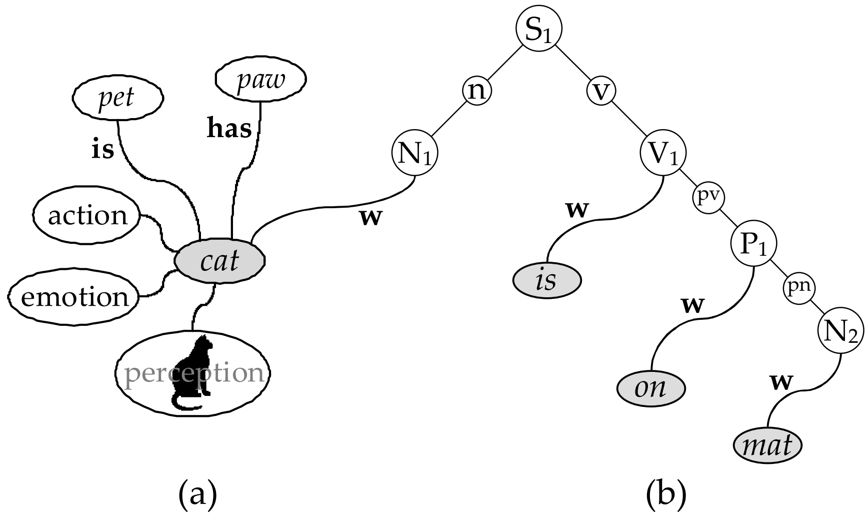

Fig. 2. In (a), illustration of the grounded word structure of a word like *cat.* In (b), illustration of the grounded sentence structure of *The cat is on the mat* (ignoring *the*).

(McClelland & Rogers, 2003; Feldman, 2006; O'Reilly, 2006; van der Velde & de Kamps, 2006). They can be implemented by specific neural circuits, e.g., such as circuits based on disinhibition (O'Reilly, 2006; van der Velde & de Kamps, 2006), or by conjunctive connections (McClelland & Rogers, 2003), or by specific activation rules (Feldman, 2006). Conditional forms of activation have been found in brain studies of rule behavior in monkeys (e.g., Miller, 1999).

The other grey ovals in figure 2b, labeled *is*, *on*, and *mat*, belong to the grounded network structures for the words *is*, *on*, and *mat*, respectively.

3. Architecture of grounded language structures

To represent a sentence, the network structures for the words (or word structures, for short) are (temporarily) bound to neural syntax structures that represent elements of syntactic information. In figure 2b these are sentence (S_1), noun phrase (N_1 and N_2), verb phrase (V_1), and prepositional phrase (P_1). The syntax structures are then bound to each other in agreement with the structure of the sentence (van der Velde & de Kamps, 2006). Binding is achieved by reverberating (delay) activity, which is neural activity that persists for a while, even when the stimulus that initiated the activity is gone. The delay activity creates a conditional connection, by which activation can flow between the bound structures. Because the delay activity decays after a while, the binding decays as well, although it can be transferred to long-term memory under certain conditions (van der Velde & de Kamps, 2006).

Figure 2 illustrates the difference between grounded cognition and symbol manipulation. Here, no symbol tokens are copied, transported or retrieved, and pasted to create sentence structures. Instead, the neural architecture for grounded language processing consists of specific connection structures and selective processes of activation. The sentence structure in figure 2 interconnects the word structures for *cat* and *mat* in such a way that the query “Where is the cat?” produces the activation of *mat*, and the query “Who is on the mat?” produces the activation of *cat*. The architecture can also represent the sentence *The cat is on the cat* (or *The mat is on the mat*). In the sentence *The cat is on the cat* the word structure for *cat* is again not copied. Instead, the same grounded word structure would be used (the one in figure 2a), connected (bound) to N_1 and N_2 in figure 2b.

Figure 2 illustrates that grounded representations are always accessible or visible, even when they are a part of a representation of a combinatorial structure, such as a sentence. Because the representation of *cat* remains visible in the structure of a sentence like *The cat is on the mat*, the connection structure of *cat* can be activated when *cat* is part of a sentence. In this way, the word structure for *cat* can influence the binding process that produces the sentence structure, or the word *cat* can be produced, e.g., in pronouncing the sentence. Furthermore, new information given by the sentence, e.g., an association between a cat and a mat, can be integrated with the word structures for *cat* and *mat*. The visibility of grounded word representations in this architecture is in agreement with the observations of Tettamanti et al. (2005) discussed above. The visible nature of grounded representations entails that they are not encoded (encrypted) in combinatorial structures. Figure 2 illustrates that accessibility (visibility) is a necessary requirement for a representation to be grounded. But it is not sufficient. Symbol tokens in a symbol manipulation architecture are accessible (visible), but they are not grounded. A representation is grounded when it is accessible (visible) and remains “in situ”, that is, when its embedded information structure always remains intact.

3.1. Binding in the architecture

In the architecture discussed here (van der Velde & de Kamps, 2006), sentence structures are created in a neural blackboard. Figure 3 illustrates basic aspects of the neural blackboard architecture. Figure 3a shows a more elaborate representation of the structure *cat sees dog*. It shows that the words are connected by memory circuits to structure assemblies for noun phrase (N_1 and N_2) or verb phase (V_1). Memory circuits are gating circuits based on disinhibition. A memory circuit is activated when the neural structures it connects (e.g., *cat* and N_1) have been active simultaneously. When a memory circuit is active, it remains active for a while (due to reverberating activity in the circuit), and it forms a temporal link between the neural structures it connects.

To form a sentence structure, structure assemblies are connected to each other. This makes the difference with a semantic network, in which semantic relations are formed between word representations directly. The intervening structure assemblies allow novel combinations between words (e.g., nouns and verbs) to be made, because the nouns are connected only to N_x assemblies (between 10 and 100) and verbs are connected only to a similar number of V_i assemblies, and the N_x and V_i assemblies are connected to each other.

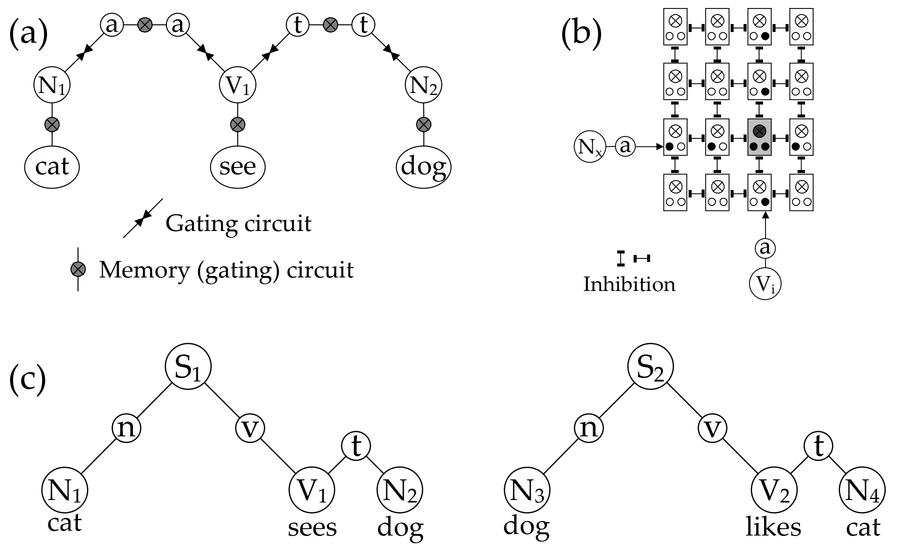

Fig. 3. (a) Basic sentence structure with explicit gating and memory circuits. (b) Connection matrix of memory circuits. (c) Multiple sentence structures (in 'shorthand' illustration).

A structure assembly consists of a main assembly (e.g., N_1 or V_1 in figure 3a) connected to a number of subassemblies. For example, V_1 has (among others) subassemblies for agent (a) and theme (t). The connection between a main assembly and a subassembly consists of a gating circuit based on disinhibition. The gating circuit can be opened by a control signal. In this way, activation flow within a structure assembly is controlled. A main assembly can be active without any of its subassemblies. Main assemblies remain active for a while, unless they are inhibited. Due to competition, only one main assembly of the same kind is active at a given moment (the last one activated, see van der Velde & de Kamps, 2006).

The structure assemblies (e.g., N_x and V_i) are interconnected by means of subassemblies of the same kind. For example, all N_x and V_i assemblies are interconnected by agent subassemblies (and by theme subassemblies). Figure 3b illustrates that subassemblies of the same kind form a connection matrix, in which each element consists of a memory circuit. When the agent subassembly of N_x is activated, it activates its row in the connection matrix. The row remains active until a binding occurs. When the agent subassembly of V_i is activated, it also activates its row in the connection matrix, which remains active until a binding occurs. When the N_x row and the V_i row in the connection matrix are active simultaneously, they activate their corresponding memory circuit. This forms a (temporal) binding between their agent subassemblies (and terminates the activation in the corresponding rows in the connection matrix). In figure 3a, N_1 and V_1 are bound by their agent subassemblies, and V_1 and N_2 are bound by their theme subassemblies, which makes *cat* the agent of *sees* and *dog* its theme.

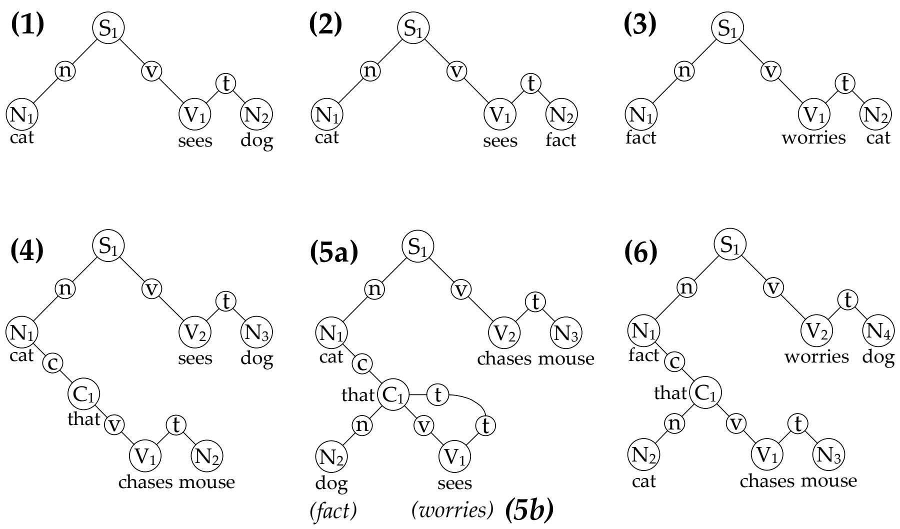

Fig. 4. Illustration of the sentence structures used in training control of binding. The first set consisted of noun-verb-noun sentences (1-3). The second set consisted of these sentences and sentences with embedded clauses (1-6).

Figure 3c shows that multiple sentences can be represented at the same time, with the same words in different roles. The sentence structures are presented in this figure in a 'shorthand' representation by omitting the gating and memory circuits. But these circuits, and their connection matrices, are always implied. Furthermore, the sentence structures also have sentence assemblies (S_1 or S_2), which provides the possibility of a more elaborate grammatical structure. Due to the structure assemblies, the same grounded word representation *cat* can be the subject of one sentence and the theme of another. The word *cat* is the subject of *cat sees dog*, because it is bound to N_1, which is bound to S_1 (by their noun subassemblies), and it is the theme of *dog likes cat*, because it is bound to N_4, which is bound to V_2 (by their theme subassemblies). With these sentences, it is possible to reliably answer queries like "Who sees the dog?" or "Who likes the cat?". Thus, words can be bound to different sentences in different roles, and different sentences can be represented simultaneously, by using different structure assemblies (for the remainder of the paper, we ignore the use of different structure assemblies to represent different sentences).

Figure 4 shows that the architecture can also represent more complex sentence structures, containing embedded sentences. We used the sentences illustrated in this figure to study how control of binding can be learned in our architecture.

4. Binding and control

The binding process in the architecture proceeds in two stages. First, a word is bound to an (arbitrary) structure assembly of its kind when the word is heard or seen. A process of how this could occur is discussed in (van der Velde & de Kamps, 2006). Furthermore, we assume that an (arbitrary) sentence assembly is activated at the beginning of a sentence. In the second stage, the structure assemblies are bound by their common-type subassemblies. This is the most important stage, because it determines the grammatical structure of a sentence representation. In (van der Velde & de Kamps, 2006) we presented a neural control circuit, as an example of how this binding process could proceed. However, the circuit was entirely hand-made, and designed for one type of sentence only.

In our present study, we trained a (three-layer) SRN and a three-layer feedforward network (FFN) to control the binding of a set of (training) sentences, and investigated how binding in the architecture would operate for another set of test sentences. The training sentences are presented in figure 4. The SRN and FFN used to control the binding process consisted of the 10 input nodes, 10 hidden nodes and 13 output nodes. They were trained using the backpropagation training procedure in (Plunkett & Elman, 1997).

Figure 5 shows our attempts to learn control of binding in the architecture using an SRN. Initially, we trained an SRN on word information only (figure 5a). That is, for a sentence like *cat sees dog*, it received the word type information *noun*, *verb* and *noun*, and it was trained to produce the correct binding signals (see below). With this procedure, training succeeded for the sentence types 1-3 in figure 4. It also succeeded for sentences of type 4. But the distinction between sentence types 4 and 6 was difficult to learn. The difficulty remained when we made an explicit distinction in the input between nouns with a (potential) relative clause, like *cat*, and nouns with a (potential) complement clause, like *fact*.

In the next step, we used feedback information from the blackboard (figure 5b). The verb *sees*, for example, could anticipate a theme (object) by activating its theme subassembly (and thus the corresponding row in the theme connection matrix). Feedback from this matrix could be used as additional input, to inform the SRN that the output should be a binding of a noun as theme. Training improved with this kind of feedback, but difficulties of binding remained.

Figure 5c illustrates the information we finally used to train the SRN (and FFN) in our study. It consists of three kinds: word type information, direct feedback from the blackboard, and conditional feedback. In the case of conditional feedback, the feedback from the blackboard activates a ‘conditional node’ that, in conjunction with specific word type information, provides input to the SRN. So, the conditional node T (theme), activated by the theme connection matrix, operates in conjunction with a noun to activate an input node of the SRN. This provides direct information that the noun should be bound as a theme. We assume that these conjunctive effects between input and conditional nodes exists, but they could result from a basic form of association. We assume that the activation of conditional nodes end once the condition they reflect is fulfilled (e.g., by feedback from the activated input nodes).

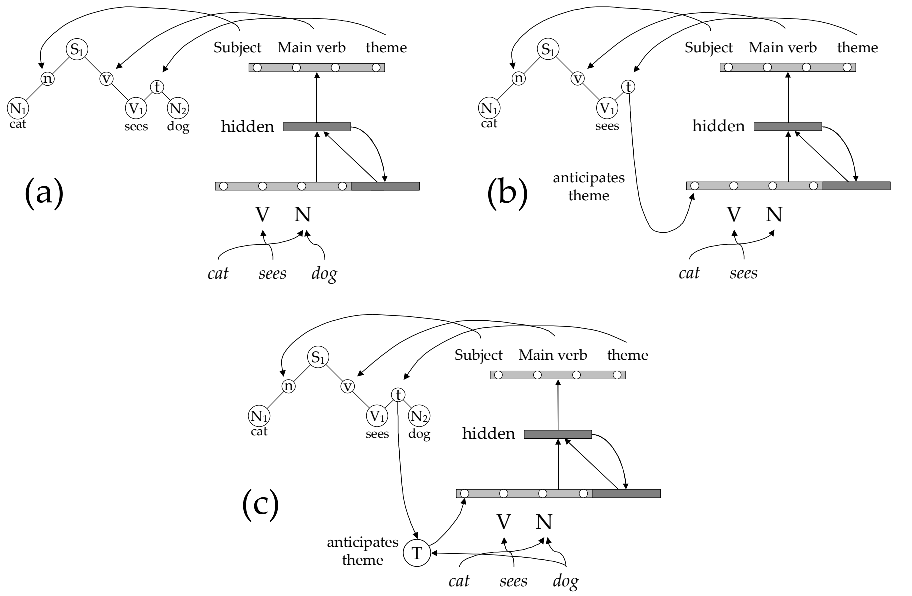

Fig. 5. Processing a sentence using an SRN for binding. (a) Without feedback. (b) With feedback. (c) With conditional feedback.

4.1. Training input-output relations

We discuss the input-output relations that the SRN (and FFN) had to learn in a step-by-step manner, based on the sentence types illustrated in figure 4. First, the SRN learned the relation between an input node and an output node signalling the beginning of a sentence, and the relation between an input node and an output node signalling the end of a sentence.

For sentence 1, the (trained) binding process proceeds as follows. The first word *cat* activates an input node (‘N_{rc}’) of the SRN representing that *cat* is a noun that can have a relative clause. For this input, the SRN must learn to activate a node representing that *cat* (N_1) should bind to S_1 as its subject. The connection matrix in figure 3b shows that this binding process consists of activating the noun subassemblies of N_1 and S_1, which activates their corresponding memory circuit. The SRN also learns to activate an output node that represents ‘relative clause’ (RC). In turn, this node activates a conditional node RC.

The verb *sees* activates a ‘verb’ (V) node in the input layer of the SRN. The output consists of the activation of two nodes. The first one initiates the binding of V_1 and S_1 by their verb subassemblies. This binding process proceeds in the same way as the binding of N_1 and S_1 discussed above. The second one (‘V-t’) activates the theme subassemblies of all V assemblies (by opening the gating circuit for these subassemblies). The effect is that the theme subassembly of V_1 is activated, which in turn activates the corresponding row in the connection matrix for theme subassemblies. This row remains active until a binding has occurred. The activation of a row in the connection matrix for theme subassemblies produces feedback by activating a conditional node T.

Finally, the word *dog* activates its word type node N_{rc}, but in conjunction with the active conditional node T it also activates an input node T. The (trained) output is the activation of the RC node (because of N_{rc}) and the activation of a node N-t that initiates the activation of the theme subassemblies of all N assemblies. In this case, the theme subassembly of N_2 is activated. In turn, it activates its row in the connection matrix for theme subassemblies. Because the row of V_1 is still active, the effect is the binding of V_1 and N_2 by their theme subassemblies. Furthermore, the activation in the connection matrix (in the rows of V_1 and N_2) is terminated, so that the conditional node T no longer receives feedback.

In sentences 2 and 3, the word *fact* activates an input node N_{cc} that represents that nouns like *fact* can have a (potential) complement clause. This creates a new output node CC and a new conditional node CC. They play the same role as the corresponding RC nodes in sentence 1.

In sentence 4, *cat* is the subject of both the main sentence and the relative clause. The binding of *cat* proceeds as in sentence 1. The word *that* activates a new input node C (clause) of the SRN. Furthermore, the combination of *that* and the conditional node RC (activated by *cat* earlier) activates a new input node 'RC'. The first (trained) output activation is that of an output node that controls the binding of N_1 and C_1 by their clause subassemblies. The second output node 'C-v' activates the verb subassembly of C_1, and thus its row in the connection matrix for verb subassemblies. The feedback of this connection matrix activates a new conditional node 'Cv' (this process is similar to the one for T illustrated in figure 5c). The third activated output node is RC, which continues the activation of the conditional node RC (without this new feedback, the activation of the conditional node RC would terminate after the activation of the input node RC).

The verb *sees* of sentence 4 activates the V input node, and, in conjunction with the conditional node Cv, activates a (new) input node 'Cv'. Two output nodes are activated. The first one, 'V-v', activates the verb subassemblies of the V assemblies, thus the verb subassembly of V_1 in this case. In turn, this produces the activation of the V_1 row in the connection matrix for verb subassemblies. In combination with the active row produced by C_1, this results in the binding of C_1 and V_1 by their verb subassemblies (and the termination of the feedback to the Cv condition node). The second output node, V-t, produces the effects discussed for sentence 1. The binding of *mouse* and *sees dog* also proceeds as discussed for sentence 1.

In sentence 5a (*cat that dog sees chases mouse*), *cat* is the subject of the main sentence but the theme (object) of the relative clause. The binding of *cat that* proceeds as in sentence 4. The word *dog* activates the N_{rc} input node (as always), but in combination with the active condition node RC (activated by the binding of *that*) it also activates the RC input node. The (trained) activated output nodes consist of a node that produces the binding of N_2 and C_1 by their noun assemblies, and the node 'C-t'. This node produces the activation of the theme subassembly of C_1, and thus of the corresponding row in the connection matrix of theme subassemblies (in this way, *cat* can be bound as the theme of the clause verb when *sees* appears). The binding of *sees* and *chases mouse* proceeds as outlined above.

The sentence *cat that the fact worries chases the mouse* (sentence 5b) has a structure comparable to sentence 5a, with *fact* in the role of *dog*. This creates the new input-output relation between the inputs N_{cc} and RC (activated by *that*) and the output node for binding N_2 and C_1 by their noun subassemblies and the output node C-t. The binding of *worries* and *chases mouse* proceeds as outlined above.

Sentence 6 introduces the (final) two new input-output relations, needed to control the binding of the complement clause in *fact that cat chases mouse worries dog*. The binding of *fact* proceeds as discussed above. The word *that* activates the input node C. In combination with the conditional node CC (activated by *fact* earlier) the word *that* also activates a new input node ‘CC’. The first (trained) output activation is that of the node that controls the binding process of N_1 and C_1 by their clause subassemblies, as before. The second (trained) output node C-v activates the verb subassembly of C_1, and thus its row in the connection matrix for verb subassemblies. The feedback of this connection matrix activates the conditional node Cv. The third (trained) output node is the node CC, which continuous the activation of the conditional node CC.

The word *cat* in sentence 6 activates the N_{rc} input node. In combination with the active condition node CC (activated by *that*), it also activates the CC input node. The (trained) activated output nodes are the one that produces the binding of N_2 and C_1 by their noun assemblies, and the node RC. The binding of *chases mouse* and *worries dog* proceeds as outlined above.

The SRN (and FFN) was trained to reproduce the input-output relations discussed above. First, the sentences 1-3 were trained (50.000 epochs), then all sentences (1-6) were trained (again 50.000 epochs). The activation of the output nodes ranged from 0 to 1 (by definition). The results showed that the SRN and the FFN produced all correct output patterns with activations > 0.9, whereas the activation of all other nodes was < 0.1 (i.e., there were no ‘spurious’ activations of output nodes). So, both networks succeeded in reproducing the input-output relations that control the binding process of the sentence types 1-6.

5. Performance on test sentences

We tested the ability of the SRN (and FFN) to control the binding process in the architecture with the set of sentence types in which the structures occurring in the training sentences were extended and recombined. In all test sentences, the FFN produced the correct output with activation > 0.9 (without spurious activations). Therefore, we concentrate our analysis of the test sentences on the performance of the SRN.

Figure 6a shows a test sentence with a multiple embedded relative clause of the kind in sentence 4 (a subject-relative clause). In this case, the theme of the first relative clause (*cat that chases*) is itself a relative clause, with *mouse* as its subject. Humans have no problem in processing sentences of this kind (Gibson, 1998). But the SRN faced a difficulty in binding the second relative clause. The signal to bind N_2 and C_2 was barely above 0.5 (about half of the desired output). If we take an

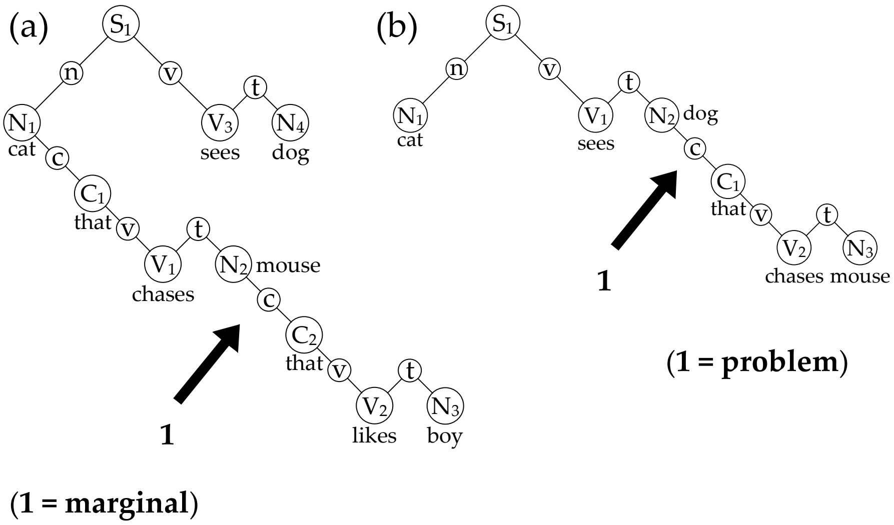

Fig. 6. Illustration of the sentence structures used in testing the control of binding based on the SRN. (Marginal: output activation ≈ 0.5. Problem: output activation < 0.5).

activation of 0.5 as a threshold of a successful binding, the binding produced by the SRN was (marginally) successful. The sequence of two clauses did not occur in the training set. It did not trouble the FFN, but the SRN apparently learned to expect particular sentence sequence, which interfered with the new sentence structure

In figure 6b the relative clause is moved from the subject position to the object (theme) position. This configuration did not occur in the training sentences. The SRN had a problem (activation < 0.5) with binding the relative clause to the theme noun. In the training sentences, it learned to produce the EoS ('End-of-Sentence') signal at this position. Most likely, this introduced a conflict when the clause appeared instead.

Figure 7a shows a sentence with a multiple subject-relative clause at the theme (object) position. The SRN has problems in binding both relative clauses. Yet, it succeeds to bind the nouns and verbs within the relative clauses. This suggest that the SRN has a small 'window of attention' for a word sequence. It gets confused when it is confronted with a new sequence, like the introduction of a clause at the theme position. But within the clause it recovers, because its has learned that (sub)sequence in the training set.

Figure 7b shows a sentence with an object-relative clause at the theme position. The first problem is again the introduction of a clause at the new theme position. The second problem consists of failing to recognize the EoS input (i.e., a failure to produce the EoS signal). The reason is most likely the fact that this sentence ends with a verb (*sees*), whereas none of the training sentences ended with a verb.

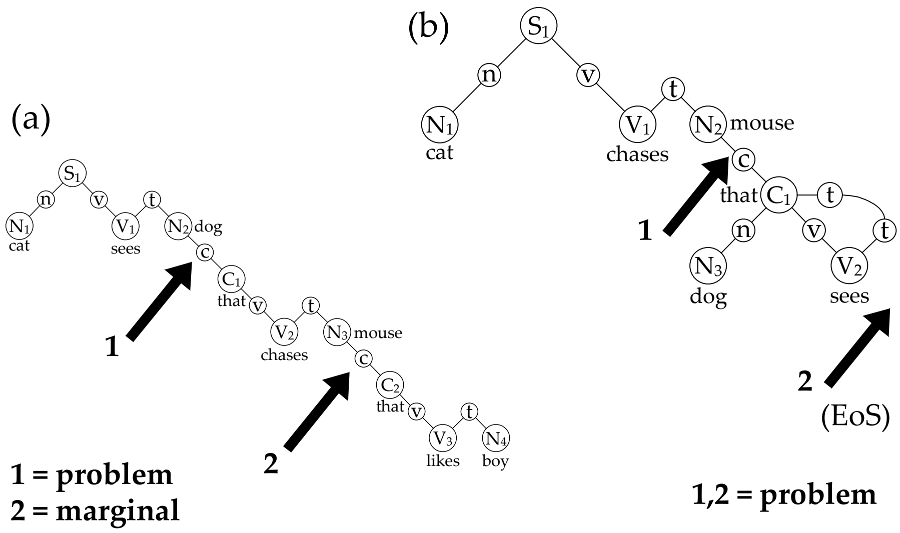

Fig. 7. Continued illustration of the sentence structures used in testing the control of binding based on the SRN. Marginal: output activation ≈ 0.5. Problem: output activation < 0.5. EoS = End of Sentence.

Figure 8a shows a sentence with a multiple object-relative clause. The SRN produced the correct binding output for this sentence, on the assumption that the blackboard produced the correct feedback. This feedback, however, is dependent on the binding process in the blackboard. In (van der Velde & de Kamps, 2006) we showed that ‘binding conflicts’ arise with sentences of this type. In particular, the verb and theme subassemblies of C_1 have been activated when the second clause (*dog that boy likes*) appears. This second clause then activates the verb and theme subassemblies of C_2. So, when the first verb *likes* appears it can bind to either C_1 or C_2 (and differently for verb or theme). The same problem arises for the other two verbs. Only when the subassemblies of C_2 are (substantially) higher activated than those of C_1 will *likes* bind to C_2 and *sees* bind to C_1. But fluctuations in the activation levels of the subassemblies involved can result in erroneous bindings (or no binding at all). Humans indeed have severe problems with sentences of this kind (Gibson, 1998).

Nevertheless, the SRN (and the FFN) produced the correct binding signals when the binding feedback produced by the blackboard is correct, that is when the binding in the blackboard succeeds without conflict. The latter would occur in an idealized situation when the level of activation of the subassemblies involved is strictly dependent on the moment of activation (i.e., when a more recently activated subassembly has a higher state of activation than a previously activated subassembly). This perhaps suggests that the overall architecture (blackboard and SRN/FFN combined) possesses the ‘competence’ to handle (arbitrary) recursive structures of this kind. The performance difficulties then arise from the noisy non-idealized dynamics in the architecture. However, the competence of the system is not embodied in the SRN/FFN alone. Instead, the input-output relations learned by the SRN/FFN

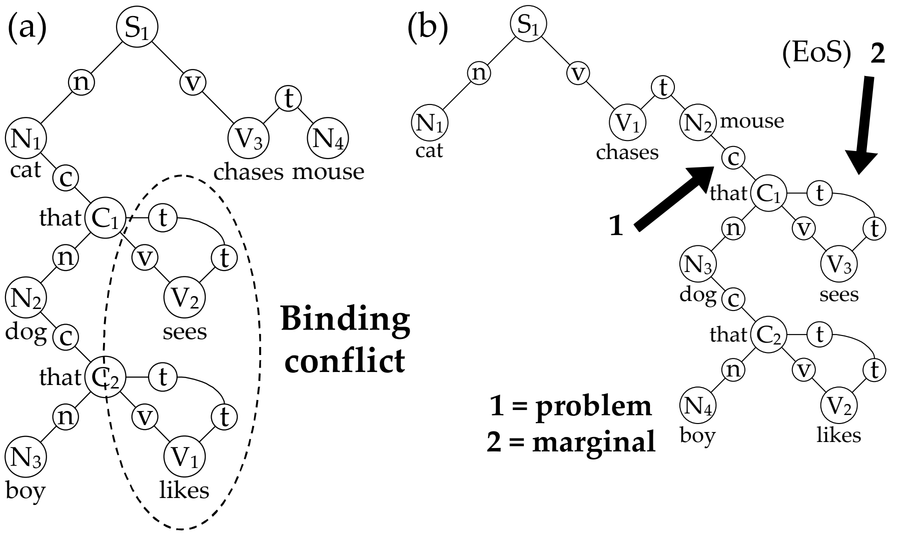

Fig. 8. Continued illustration of the sentence structures used in testing the control of binding based on the SRN. Marginal: output activation ≈ 0.5. Problem: output activation < 0.5. EoS = End of Sentence.

suffice because a substantial part of sentence structure is embodied in the architecture itself. The feedback from the architecture is thus an integrated part of its competence, which indicates that competence and performance are integrated in this architecture.

Figure 8b shows the multiple object-relative clause moved to the object (theme) position. The SRN failed to produce the correct binding signal for the binding of the relative clause to the object (as in previous examples), and it marginally produced the EoS signal. But it produced the correct binding signals within the clauses, which emphasizes the small 'windows of attention' analysis given above. Notice that a binding conflict would also arise between C_1 and C_2 with this sentence, although the fact that only two verbs are involved makes this binding conflict less severe than the one in figure 8a.

Figure 9a presents a multiple embedded clause consisting of a complement clause (CC) followed by a relative clause (RC). The SRN produced the correct binding output, based on the correct feedback from the blackboard. However, the binding output for the verb in the CC clause was marginal. For this sentence, there is only one binding conflict (between the verb subassemblies of C_1 and C_2), instead of two, as in figure 8. This makes this sentence less complex than the one in figure 8a, in line with human performance (Gibson, 1998).

Figure 9b shows the result for the transition of the CC-RC structure to the object position. This creates the usual difficulty of binding the first clause to the object (theme) noun. It also repeats the marginal binding of the verb in the CC. An additional problem arises for binding the noun in the CC. This problem also occurred when the CC was moved to the object position without the RC. We have not shown that simulation result here, because of its replication in figure 9b. It is not quite clear

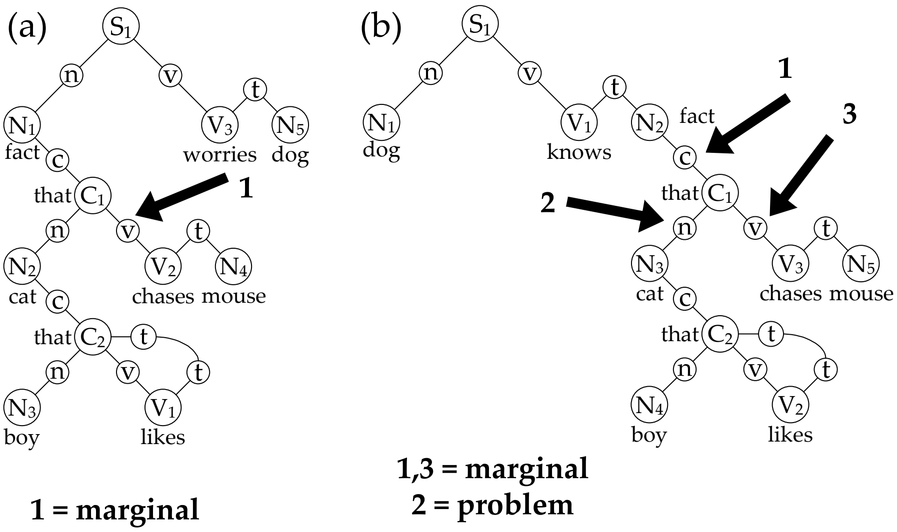

Fig. 9. Continued illustration of the sentence structures used in testing the control of binding based on the SRN. Marginal: output activation ≈ 0.5. Problem: output activation < 0.5.

why a transition of a CC to the object position is harder for the SRN than a transition of a RC to that position.

Figure 10a presents a multiple embedded clause consisting of a relative clause

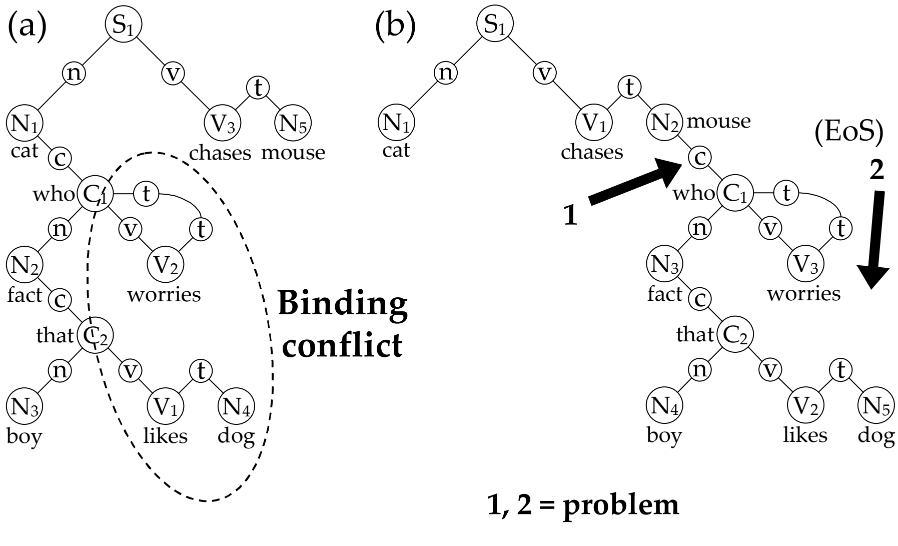

Fig. 10. Continued illustration of the sentence structures used in testing the control of binding based on the SRN. Marginal: output activation ≈ 0.5. Problem: output activation < 0.5. EoS = End of Sentence.

followed by a complement clause. These RC-CC sentences are very hard for humans to process (Gibson, 1998). Again, the SRN produced the correct output (binding signals) when feedback from the blackboard was correct. But there are two binding conflicts with this sentence, one consisting of the verb subassemblies of C_1 and C_2, and one consisting of the theme subassemblies of C_1 and N_4. These binding conflicts are of a similar kind compared to the sentence in figure 8a. The RC-CC sentence in figure 10 causes processing difficulties for humans that are similar to those of the sentence in figure 8a (Gibson, 1998). Finally, the transition of the RC-CC clause to the object position causes the usual problems of binding the clause to the object (theme) noun, and producing the EoS (end of sentence) signal after a verb.

6. Conclusions and discussion

We discussed a neural architecture that aims to integrate three important features of human cognition: productivity, dynamics and grounding. Productivity refers to the combinatorial nature of cognition, as found in language, reasoning and vision. Dynamics refers to the ability to interact with the environment in a dynamical way. Grounding refers to the nature of cognitive representations: representations for concepts are always grounded in perception, action, emotion, and associations, and embedded in semantic relations and other cognitive structures.

Individually, each of these features is important for cognition. But, in particular, their combination is important. Productivity is found in computer systems, but there it is unlimited, and not dynamic and grounded. Dynamical systems are abound in nature, but most of them are not cognitive. Grounded representations form the backbone of associative processing in all neural systems, but associative processing is not productive. So, the combination of productivity and grounding, instantiated in a dynamical system, is rare. The human brain is perhaps the only known system of this kind. An architecture that integrates these features is therefore important for understanding the nature of the brain, and for understanding the nature of human-like cognition.

The combination of productivity, dynamics and grounding is in particular importance because of the constraints they impose on each other. In turn, these constraints can reveal fundamental aspects of human cognition. For example, combinatorial structures can be created with grounded representations, but not all structures are equally feasible, as the examples illustrated in figures 8 and 10 show.

In this paper, we discussed the issue of learning in particular. The examples given are basic, and the learning mechanism is not a learning mechanism of natural language by far. Yet, even the basic examples given demonstrate how the issue of language learning could be approached. We used an SRN and an FFN to learn input-output relations to control the binding process in the architecture. The FFN produced all binding relations successfully, both for the training sentences and the test sentences.

In contrast, binding with the SRN resulted in a number of problems and marginal binding signals. In particular, these problems arose when familiar structures like clauses occurred at novel positions, not seen in the training examples, or when a verb occurred at the end of a sentence, which also not occurred in the training sentences. These problems again show that the ability of SRNs to store sequences of words interfere with the combinatorial productivity of language (van der Velde et al., 2004). Not all sequences of words that can appear in a sentence context can be learned in a lifetime. Therefore, an architecture for processing language should be flexible, and not be led astray by a novel sequence of words, not seen before.

The FFN was able to handle the more complex and recombined sentence types. However, as we noted above, the input-output relations learned by the FFN suffice because a substantial part of sentence structure is embodied in the architecture itself. This shows that feedback from the architecture is an integrated part of the competence of this architecture. That is, control of binding in this architecture is based on a recurrent network, of which the trained FFN is just a part.

The involvement of the entire architecture in control of binding raises the question of how the structure of the architecture itself emerges. We argue that this could be the result of a different kind of process that might be referred to as ‘development’ instead of just learning. The SRN and FFN we used show what could be the difference between learning and development. The SRN and FFN learn by an adaptation of their weights, but their structure does not change. Perhaps one could introduce a structure change by allowing new connections to be formed in the networks. But even then, the structure change is not on a par with that needed to develop the structure of the blackboard architecture (i.e., the different assemblies and subassemblies involved, and their corresponding connection matrices). The success of learning with the FFN perhaps suggest that language learning is the result of such a dual process: an adaptation of control, in combination with a structural development of a binding architecture for grounded representations.

This distinction might motivate the search for two different kinds of mechanisms: one needed for the structural development of a language architecture; and one needed for control, based on feedback from this architecture. The first could be referred to as ‘development’, because it is primarily related to structural changes that might occur only at an initial stage. The latter could be referred to as learning, based on continuously updated information. This dual strategy might be more successful than the search for a single learning mechanism that produces language learning from scratch. The elaboration of this strategy is one of the future developments of the architecture presented here.

References

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning, 7,* 195-225.

Feldman, J. A. (2006). *From Molecule to Metaphor*. Cambridge, MA: MIT Press.

Gibson, E (1998). Linguistic complexity: Locality of syntactic dependencies, *Cognition, 68,* 1-76.

McClelland, J. L., & T. Rogers, T. (2003). The parallel distributed processing approach to semantic cognition, *Nature Reviews Neuroscience, 4,* 310-322.

Miller, E. K.(1999). The prefrontal cortex: complex neural properties for complex behavior, *Neuron, 22,* 15-17.

O'Reilly, R. C. (2006). Biologically based computational models of high-level cognition, *Science, 314*, 91-94.

Plunkett, K. & Elman, J. L. (1997). *Exercises in rethinking innateness*. Cambridge, MA: MIT Press.

Tettamanti et al. (2005). Listening to action-related sentences activates fronto-parietal motor circuits, *Journal of Cognitive Neuroscience, 17*, 273-281.

van der Velde, F. van der Kleij van der Voort. G. T. & de Kamps, M. (2004). Lack of combinatorial productivity in language processing with simple recurrent networks. *Connection Science, 16*, 21-46.

van der Velde. F., & de Kamps, M. (2006). Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, *22*, 37-70.

Vigliocco, G., Warren, J., Siri, S., Arciuli, J. Scott, S. & R. Wise (2006). The role of semantics and grammatical class in the neural representation of words, *Cerebral Cortex, 16*, 1790-1796.