A Split Model to Deal with Semantic Anomalies in the Task of Word Prediction

Janet Hui-wen Hsiao (h.hsiao@sms.ed.ac.uk)
Division of Informatics, University of Edinburgh
2 Buccleuch Place, Edinburgh, EH8 9LW, UK

Abstract

On the task of predicting the range of possible next words in a sentence, many networks (e.g. Elman, 1990) that have been proposed are capable of displaying a certain degree of systematicity, but fail in recognizing grammatically correct but semantically anomalous sentences. Based on an expansion of Hadley's model (Hadley et al, 2001), I present a competitive network, which employs two sub-networks that discern coarse-grained and fine-grained categories respectively, by being trained via different parameter settings. Hence, one of the sub-networks will have a greater capacity for recognizing the syntactic structure of the preceding words, while the other will have a greater capacity for recognizing the semantic structure. This corresponds to the recent suggestion about specialization of the two hemispheres in the human brain (Beeman, 1998). Also, a mechanism to switch attention between the predictions from the two sub-networks is employed in order to make the global network more closely approximate human behavior. The results show that the network is able to deal with grammatically correct but semantically anomalous sentences.

Introduction

Since 1990, several cognitive scientists have concentrated on the capacity of connectionist networks to display systematicity in the task of predicting the range of possible next words in a sentence (e.g. Elman, 1990, 1998; Christiansen and Chater, 1994; Hadley, 1994a, 1994b, 2001; Marcus, 1998; Phillips, 2000)[1]. Some also proposed networks which are able to discover hierarchical semantic categories and predict according to the semantic constraints they have acquired from contexts (e.g. Elman, 1990). However, the issue of the networks' response to semantically anomalous sentences was barely addressed. By definition, semantic constraints are the semantic patterns we habitually encounter, and a semantically anomalous sentence is a sentence containing semantic patterns that violate the semantic nature of these semantic constraints. An example of semantically anomalous sentences is "boys eat rocks", which violates the semantic constraint that the word following "eat" should be an edible noun. The task here is to deal with a subset of such sentences.

Ideally, the connectionist network is expected to make syntactic predictions, instead of semantic predictions, for grammatically correct but semantically anomalous sentences. If we train the network to generalize the input words more and recognize fewer subcategories, it may not have the capacity to discover all the semantic constraints. On the other hand, if we train the network to recognize more subcategories, it will lose the information about general categories to make syntactic predictions. Therefore, to train the network with suitable parameters, which will enable the network to handle both situations well, is a challenging task.

Based on an expansion of Hadley's work (2001), a more challenging training corpus is created according to a set of semantic constraints. It is believed that humans require both semantic and syntactic information to deal with semantic anomalies. So that when we encounter a semantically anomalous sentence such as "boys eat rocks", we can still recognize it as a grammatical sentence. Hence, a mechanism to learn information from both general categories (e.g., noun) and subcategories (e.g., human noun) is required in the network design. A way to achieve this is to use two sub-networks which respectively learn information about categories and subcategories, by using different training parameter settings. It is assumed that a network can recognize a grammatical sentence if a period is predicted at the right place and if it can make predictions according to correct English grammar. A failure to make substantial semantic predictions suggests that current input contains a novel semantic pattern that the network does not habitually encounter during training, i.e., a semantically anomalous sentence. Moreover, during testing, a mechanism to coordinate the information exchange between the two sub-networks is used in the hope that it will help the network make predictions close to human behavior.

System Overview

The task of the network proposed here is to learn to predict semantic features of the next word, given a prior sequence of words. Words are taken from a pool of sentences generated according to a simple syntax displayed in Figure 1.

[1] Or rather, we could say that the task of the network is to "anticipate" the range of possible next words in a sentence.

S -> NP V NP .
NP -> N | N RC | N PP
N -> NOUN-HUM | NOUN-ANIM | NOUN-INANIM | NOUN-FOOD
V -> VERB-EAT | VERB-PERC | VERB-TRAN | VERB-STREN | VERB-HIT
RC -> that V NP
RC -> that N V
PP -> PREP NP

Figure 1: The grammar for generating training and test sentences.

In our corpora, the vocabulary contains 16 nouns, 16 verbs, and 2 prepositions. All words have been previously assigned semantic feature vectors with 60 features taking binary values. A unit in the encoding of a word is set to one if the word exhibits the feature, and zero otherwise. Among the 60 features, 23 features are assigned to nouns, 21 are to verbs, and the remaining 16 features are reserved for the words (1) "that", (2) "with", (3) "from", and (4) the period ".", which do not have straightforward semantic information. These 16 features are divided equally and assigned to the four words above. They might be viewed as syntactic representations (Hadley et al, 2001), since these four words serve as function words, which are semantically light and used to signal structure. The creation of semantic features here is admittedly somewhat arbitrary. However, it has conveyed the general approach adopted here. That is, if semantic features do exist in the human language acquisition mechanism, the proposed network is able to provide a possible computational model for dealing with semantic anomalies.

The sentences in the training corpus are generated according to a set of semantic constraints: all the simple sentences and all clauses of complex sentences must fall into one of the semantic structures defined in the semantic constraints. See Figure 2 for some examples.

NOUN-HUM VERB-EAT NOUN-FOOD
NOUN-HUM VERB-EAT NOUN-FOOD with NOUN-HUMAN
NOUN-HUM VERB-PERCEPT NOUN-INANIM
NOUN-HUM VERB-TRAN NOUN-HUM
…

Figure 2: Examples of semantic constraints

Figure 3 shows the network architecture. The arrows in the figure represent entire sets of links between two layers. Dotted arrows indicate trainable links. The training of the network involves only the portion inside the dotted square, referred to as the *training network*. The training network consists of four layers: an input layer, a first hidden layer (HL1), a second hidden layer (HL2), and an output layer. In short, two of Hadley's Hebbian-competitive networks (Hadley et al, 2001) are put together side by side, sharing the same input layer, and train them independently. Different parameter settings are used in the two sub-networks to make the left sub-network recognize general categories (e.g. the correct usage of English grammar) and the right sub-network recognize sub-categories (e.g. human or inanimate nouns). This also corresponds with the recent suggestion that the two hemispheres in the human brain activate different breadth of semantic fields[2] (Beeman, 1998).

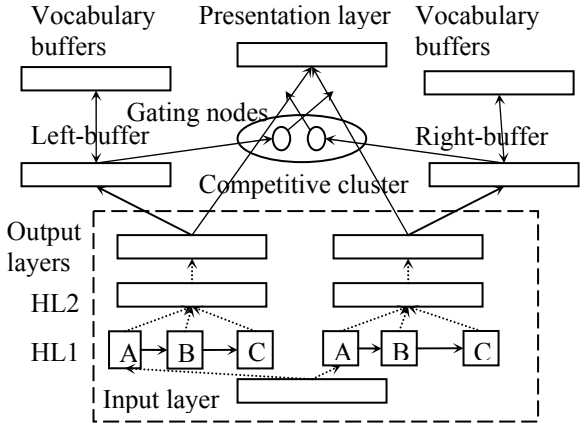

Figure 3: Network architecture.

Training Phase The training corpus contains 10,000 sentences. Half of the sentences in the training corpus are simple sentences with the form NOUN VERB NOUN. 25% of the sentences contain a single prepositional phrase. The rest sentences contain one or two relative clauses. During the training phase, 50,000 sentences were randomly selected from the training corpus and presented to the network. The two sub-networks are trained with the same algorithm and winner selection rules as used in Hadley's networks (see Hadley et al, 2001), except that the links from HL2 to the output layer are trained via a *reverse competitive learning algorithm*, described below. In short, area A has a post-training role of categorizing the input feature vectors into semantic groups. Area B and C store the previous successive contents of area A. The role of HL2 is to be a higher order pattern recognizer to categorize the ternary patterns that appear in the three areas of HL1. The output layer receives activation from HL2 and is trained to make semantic predictions of the next word.

Also, the sum of the weights on all the links from each node in HL2 to the output layer is set to one, so

[2] Taken from Beeman's explanation, "the subset of semantic information activated in response to an input word is termed the semantic field, a projective field comprising the set of internal representational units (semantic features) that are activated by an external input (a word)".

that the total activation predicted in the output layer will also sum to one. Before training, the weights are distributed evenly on the outgoing links from each node in HL2. The weight modification equation is:

$$\Delta\omega_{ij} = g\frac{c_{ik}}{n_k} - gw_{ij}$$

where i is the index of the output layer; j is the current active node in HL2; n_k is the number of active nodes in the output layer; c_{ik} is equal to one if node i in the output layer is active and 0 otherwise; g is the learning rate. Notice that the modification equation resembles the one in the original competitive learning algorithm (von der Malsburg, 1973). It is actually a *reverse competitive learning algorithm* since the input layer and the competitive cluster have been put upside down with respect to the original algorithm.

The basic idea of this algorithm is that it takes a small amount of weight, decided by the learning rate, from all links connected to inactive nodes in the output layer, and then redistributes the weight to the links connected to active nodes (see Figure 4). With this method, we can strengthen the links between two active nodes and weaken those between an active and an inactive node, while keeping the sum of weights from any given node in HL2 to the output layer to be equal to one. This method actually preserves the basic idea of Hebbian learning (Hebb, 1949). One of the advantages of using this algorithm over the simple Hebbian increment model is that the sum of activation presented in the output layer will be restricted to one, instead of being incremented without a limit. The assumption of this design is that prior to training, every semantic feature will be equally weakly predicted. If a node in HL2 never wins during training, the weights on the links from this node to the output layer will never be changed. Thus, every semantic feature in the output layer will still have been equally weakly predicted.

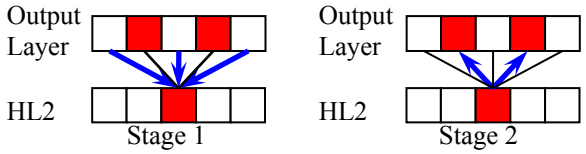

Figure 4: Weight modification in the reverse competitive learning algorithm.

The two sub-networks in the proposed network are trained independently with different parameter settings, i.e. the learning rate and the constant c in the winner selection rule (see Hadley et al, 2001). They then develop different weight configurations. A larger learning rate, 0.5, and a larger constant c, 1.0, are used in the area A of the left sub-network, to make it form groups of general categories, and a smaller learning rate, 0.02, and a smaller constant c, 0.6, in the area A of the right sub-network, to make it form groups of sub-categories. (See Hadley et al, 2001 for the influence of adopting different values for the parameters.)

Test Phase During testing, the two sub-networks take the same input, go through the same process as training, except that no weight modification occurs, and input words are presented only in the input layer. Sentences involving an anomalous combination of the agent, the action, or the patient, are created to examine the capacity of the network to deal with semantic anomalies. An example of sentences with an anomalous combination of the agent and the action is "rocks eat cookies". Sentences such as "boys eat tables" involve an anomalous combination of the action and the patient.[3]

During the test phase, another mechanism to orchestrate the interaction between the predictions from both sub-networks is placed on top of the original network (see Figure 1). Besides the training network, the test network also contains vocabulary buffers to store the semantic vectors of each word in the network's vocabulary, a left-buffer to store a copy of the left output layer, a right-buffer to store a copy of the right output layer, a presentation layer to store the final semantic predictions, and a competitive cluster of two gating nodes to gate the activation from the output layers to the presentation layer. The two sub-networks will make predictions respectively and compete with each other to present a result to the presentation layer through the competitive gating-node cluster.

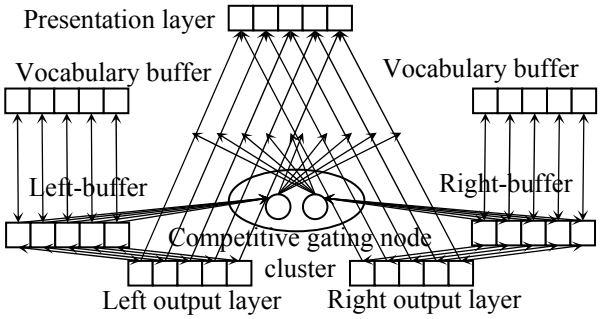

Figure 5: The detailed structure on the top of the two output layers.

In the left sub-network, links from the left output layer to the left-buffer, and from the left-buffer to the vocabulary buffer are all one-to-one copy links and

[3] Notice that a semantic anomaly, specifically defined in the models proposed here, is any sentence that violates the semantic constraints used for generating the training corpus. For experimental purposes, the semantic constraints contain some simplifying assumptions that admittedly are not always in compliance with English semantics.

have weights of +1. On the other hand, links from the left-buffer to the left gating node, and from the left gating node to the links between the right output layer and the presentation layer, are fully connected. The same applies to the links in the right sub-network. The vocabulary buffers, the left-buffer, the right-buffer, and the presentation layer all have the same size as the output layers. Each node in the presentation layer is connected to the corresponding nodes in the output layers of the two sub-networks, forming a ternary structure (see Figure 5). The left and right gating nodes receive activation from the left-buffer and the right-buffer, respectively. Each outgoing link of a gating node serves as a modifier link to inhibit the activation in the output layer of the opposite sub-network from going up to presentation layer.

When predictions from both sub-networks are activated in the lowest output layers, the two sub-networks will compete with each other according to their coherence with the network's vocabulary. Coherence is a measure of similarity between the predicted vector and the various semantic vectors in the network's vocabulary. The more the predicted vector resembles the preassigned semantic features of a certain word in the network's vocabulary, the greater the degree of coherence it has. In other words, if the predicted vector covers a broad range of semantic features, and it is hard to tell which word the vector is predicting, then that predicted vector will have less coherence. The activation level of the predicted vectors in the two output layers are boosted according to their degree of coherence. The one with greater coherence will be activated in the presentation layer. This process is called *coherence reinforcement process*, and is explained in detail below.

After predictions are activated in the output layers, the content of the left output layer is copied into the left-buffer, and the content of the right output layer is copied into the right-buffer. Each word in the network's vocabulary will be presented in the vocabulary buffer in turn. For each pair of nodes between the left-buffer and the vocabulary buffer, if either member of the pair has a value below a predetermined *reinforcement threshold*, a boost of activation will not occur. However, if both of them have values above the threshold, a boost of activation, which is proportional to the square of the activation value of the node in the left-buffer, will be added to the node in the left-buffer. Each word in the vocabulary will be activated in the vocabulary buffer in turn and go through the same process[4]. The same applies to the right-buffer in the right sub-network. After reinforcement is complete, the sub-network whose predicted vector has greater coherence with the network's vocabulary will be the one that has greater activation in total in its output layer buffer.

Recall that prior to training, every outgoing link from a node in HL2 to the output layer is given an equal fractional weight. This equal fractional weight, if not incremented during training, will later not be able to generate activation above the *reinforcement threshold* and hence will not be reinforced. In other words, if any node that has never been selected as a winning node during training is later selected as the winner in HL2, none of the features in the subsequently predicted semantic vector will be reinforced. So, if the predictions from the other sub-network have gained some reinforcement, they will be eventually activated in the presentation layer.

The competitive gating-node cluster manages predictions that will eventually be activated in the presentation layer between the two sub-networks. In the initial state, the two gating nodes have the same high-level activation and inhibit the predictions from being activated in the presentation layer. Since the two sub-networks have been trained with different parameter settings, the predictions in the output layers are also different. After the coherence reinforcement process is performed, the two gating nodes will also receive different activation values. Hence, the gating node that initially received less activation will cease its inhibition, and the predictions in the other output layer will be activated in the presentation layer. Notice that during the coherence reinforcement process, only the activation in the left-buffer and the right-buffer is reinforced. The original activation values in the output layers are still intact. Therefore, it is the original activation in the output layer of the winning network that is spread up to the presentation layer.

The predictions are evaluated by calculating the cosine value of the angle between the predicted semantic vector and the semantic vector of each word in the vocabulary. The greater the cosine value is, the closer the two vectors are, or in other words, the more strongly the given word is predicted.

Experimental Results

In the predictions of the right sub-network following an anomalous combination of agent, action, or patient in a sentence, all features have equally weak activation as their initial state. This suggests that a novel semantic-syntactic pattern that has never been seen during training is formed in HL1. Consequently, a node that has never won during training is selected as the winner in HL2, and the links from the winner to the output layer have never been trained. The equally weak activation on each feature reveals the network's inability to make semantic predictions for the given

[4] If there is indeed a process in the brain similar to the coherence reinforcement process proposed here, this process would be expected to occur in parallel.

input sentence. On the other hand, the left sub-network is still able to make predictions. Figure 6 shows predictions from the left sub-network following an anomalous sentence. It has the same distribution of predictions as the predictions for normal sentences with a pattern *"Noun Verb Noun"*. The same predictions can also be found in the presentation layer. This indicates that the left sub-network wins the competition after the coherence reinforcement process.

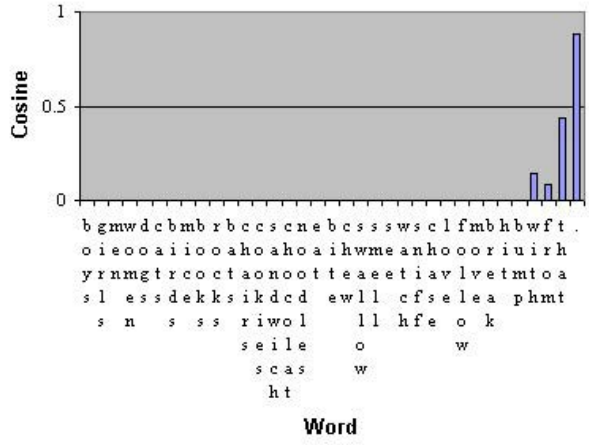

Figure 6: The predictions following a sequence of words "boys eat rocks" for the left sub-network and the presentation layer.

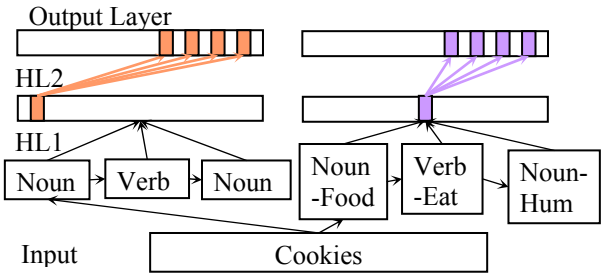

Figure 7: The predictions when the current input sentence is a normal sentence "boys eat cookies".

For example, when the current input sentence is a normal sentence, such as "boys eat cookies", (see Figure 7), and the current input word is "cookies", the category information of the first two words, "boys" and "eat", have been respectively stored in area C and B of HL1. The left sub-network recognizes "boys" as a *Noun*, and "eat" as a *Verb*, while the right sub-network recognizes "boys" as a *Noun-Human* and "Eat" as a *Verb-Eat*. When the word "cookies" comes into the input layer, the left sub-network will recognize "cookies" as a *Noun*, since it is trained to recognize general categories. On the other hand, the right sub-network will recognize "cookies" as a *Noun-Food*, since it is trained to recognize sub-categories. A winner in HL2 is then selected in each sub-network. Theoretically, these patterns should have been seen during training, so a period will be predicted by both sub-networks. Thus, for normal sentences, both networks should give good predictions.

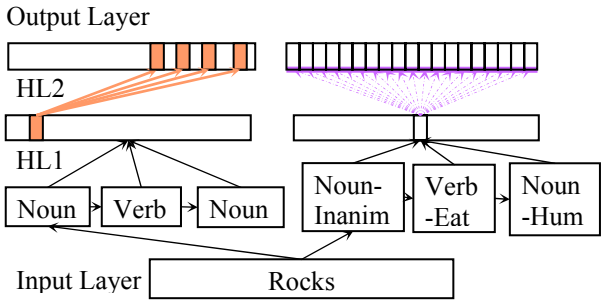

Figure 8: The predictions when the current sentence is the semantically anomalous sentence, "boys eat rocks".

When the current input is the word "rocks" in a semantically anomalous but grammatically correct sentence, such as "boys eat rocks", (see Figure 8), the left sub-network still recognizes "rocks" as a *Noun*, but the right sub-network recognizes it as a *Noun-Inanim*. Thus, in the right sub-network, an entirely new triadic pattern is formed in HL1, and a new winner will consequently be picked in HL2. Since this winner has never won during training, the weights on the links from the winner in HL2 to the output layer have never been adjusted. Hence, the right sub-network only generates an equal fractional prediction for every feature. These unsubstantial predictions indicate a semantic anomaly. On the other hand, the left sub-network still predicts a period, indicating its capacity for recognizing a grammatical sentence.

As revealed in Figure 7 and 8, when encountering a semantically anomalous sentence, the network fails to make semantic predictions, but still recognizes that it is a grammatical sentence and makes predictions accordingly. This suggests that some mechanism similar to this network might be found within a larger language acquisition system to explain how humans deal with semantically anomalies.

Discussion and Conclusion

I have presented a connectionist network which is able to deal with semantic anomalies. More specifically, for semantically anomalous but grammatically correct sentences, it fails to predict according to the anomalous semantic pattern, but it is able to predict according to their syntactic structures instead. It is the employment of two identical sub-networks, which are trained via different parameter settings during training to recognize different fineness of grains of categorization, that provides the network with the capacity for dealing with semantic anomalies. This also corresponds with the

anatomical and psychological evidence that both hemispheres are necessary for full sentence comprehension (Beeman, 1998).

Also, the network employs a *coherence reinforcement mechanism* on top of the two sub-networks to enable an information switch between them. In a separate examination, it has been found that with this mechanism, for normal sentences, most predictions from the global network are closer to what is actually presented in the test corpus, than those from any of the sub-networks alone. Thus, the network has provided a possible computational model to simulate human behavior on predicting the range of possible next words in either a normal sentence or a semantically anomalous sentence.

It is believed that making predictions for the next word in a sentence not only requires syntactic information, but also semantic information. Most previous works on the issue of systematicity have focused on the syntactic category that the network predicts (Christiansen and Chater, 1994; Elman, 1998). The authors usually trained the networks, with certain parameter settings, to be sensitive to only the syntactic structures of input patterns. However, it is probable that humans switch back and forth between semantic and syntactic information to make good predictions. For example, we require semantic information for "eat" to predict a food noun after it. On the other hand, we require syntactic information to predict the appearance of function words such as prepositions. The proposed network here is intended to draw attention to the issue of the interaction between syntactic and semantic information, and possibly between the two hemispheres, in cognitive modeling. Taking a suggestion from Hadley, the two sub-networks have been successfully trained to be sensitive to different semantic-syntactic structures, by adjusting the parameters in the winner selection rule. Also, the coherence reinforcement process successfully switches the attention between the two sub-networks. Hence, the network can more closely simulate human behavior, especially when encountering semantically anomalies.

We can further compare the network's behavior with that of human subjects through psychological experiments. However, the human brain is like a "black box" — it is difficult to understand how cognitive processes happen in the brain directly. The main source for cognitive psychologists to understand human cognition is to explore the brain indirectly through the understanding of deficient cognition. The same applies to verifications of computational models, since any computational model of human cognitive processes is useless if it cannot address psychological phenomena. Thus, to further verify and challenge the proposed network, we can examine whether it can address phenomena or explain causes of deficits in language acquisition, such as language deficits in aphasia or dyslexia.

The proposed model here is not claimed to provide a general language acquisition mechanism. The lack of biological evidence also means that we cannot be certain of a true computational model for human language acquisition processes in the brain. However, with the employment of both syntactic and semantic information, or rather, information about both fine-grained and coarse-grained syntactic-semantic categories, the proposed network has successfully provided a possible framework to deal with a subset of semantic anomalies within a connectionist network and raised the issue of the interaction between syntactic and semantic information.

Acknowledgments

Special thanks to Dr. Robert F. Hadley for his idea of using two sub-networks to acquire different information in this learning task, and valuable help throughout this research.

References

Beeman, M. (1998). Coarse semantic coding and discourse comprehension. *Right hemisphere Language Comprehension: Perspective from cognitive neuroscience.* Mahwah, NJ, UAS: Lawrence Erlbaum Associates.

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

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

Elman, J.L. (1998). Generalization, simple recurrent networks, and the emergence of structure. *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society.* Mahwah, NJ: Lawrence Erlbaum Associates.

Hadley, R.F. (1994a). Systematicity in Connectionist Language Learning, *Mind and Language,* 9(3).

Hadley, R.F. (1994b). Systematicity Revisited, *Mind and Language, 9*, 431-444.

Hadley, R.F., Rotaru-Varga, A., Arnold, D.V., Cardei, V.C. (2001). Syntactic Systematicity Arising from Semantic Predictions in a Hebbian-Competitive Network, *Connection Science*, *13*, 73-94.

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

Marcus, G. (1998). Rethinking eliminative connectionism. *Cognitive Psychology*, *37*.

Phillips, S. (2000). Constituent similarity and systematicity: The limits of first-order connectionism. *Connection Science, 12(1)*, 45-63.

von der Malsburg, C. (1973). Self-Organization of Orientation Sensitive Cell in Stirate Cortex, *Kybernetik*, *14*, pp. 85-100.