

# Overview of CALM (draft)

Adriaan Tijsseling

## Contents

Abstract.....	1
1. Introduction.....	1
2. Architecture of CALM.....	3
3. Functioning of CALM.....	6
4. Supervised learning.....	8
5. Multi-modular architectures.....	9
6. Nonlinear dynamics of multi-modular architectures.....	11
7. Self-organization with CALMMaps.....	12
8. Processing sequential information.....	15
9. Autonomous addition and pruning of nodes.....	16
Bibliography.....	19
Appendix I: Parameter values.....	22
Appendix II: Learning with negative weights.....	22

## Abstract

*This letter provides a brief overview of the current status of the Categorization and Learning Module neural network architecture, which was originally developed by Murre, Phaf, & Wolters (1992). We provide a concise introduction of the CALM algorithm and how it has been motivated by neuropsychological data. Subsequently, we discuss the design of modular architectures as well as their exhibited possibly chaotic nonlinear dynamics, which may improve categorization performance. Following this, we will describe a self-organizing variant of CALM, called CALMMap, and the addition of time-delay connections that makes processing of sequential learning possible.*

## 1. Introduction

How can a learning system be designed to remain plastic, or adaptive, in response to significant events and yet remain stable in response to irrelevant events? Plasticity is necessary for the incorporation of new representations in a network. Stability means keeping old representations intact. A solution to this trade-off is to find a mechanism that is able to distinguish between old and new representations and to use this information to control the learning process. Murre, Phaf, and Wolters (1992) have designed a neural network with this mechanism that does not suffer from problems occurring with most artificial networks, in particular non-modular networks. These problems include lack of speed, impaired stability, and an inability to learn and both discriminate between and generalize over patterns. In general, difficulties that often arise with networks that are unstructured and/or that assume total connectivity. Murre et al.'s (1992) Categorizing and Learning Module (CALM) overcomes many of these problems by incorporating structural constraints based on properties of nervous systems, such as modularity and organization with excitatory and inhibitory connections. Data from neuroscience, biology and psychology suggest that the human information processing system involves modules, relatively isolated subsystems, that can function independently of each other (Happel and Murre, 1994; Murre, Phaf, and Wolters, 1992; Murre, 1992). The right design for an initial architecture for a neural network can improve learning, because it determines what can and cannot be learned.

The idea behind CALM is that the richness of human cognitive behavior is in some way related to the modular structure of both the human brain and human information processing. The latter is characterized by relatively isolated subsystems that can function quite independently of each other. We can, for example, speak while driving a car (Shallice, McLeod, and Lewis, 1985), but this ability is lost as soon as two behaviors share the same

subsystem as when we try to lip-read while driving. Another important property of this modular organization of information processing is that isolated abilities, such as face-recognition (Damasio, Damasio, and Van Hoesen, 1982), may be lost without affecting other cognitive abilities in any way (Luria, 1973).

Structural modularity is not only found at a functional level, but also at the architectural level. The human neural system is characterized by many structurally different cortical and subcortical centers that are only partly connected. (Livingstone and Hubel, 1988; Kosslyn, Flynn, Amsterdam, and Wang, 1990). The neo-cortex itself has been found to be organized into functional columns. Each such cortical column or module forms a micro-circuit, which is characterized by a strong inhibition between neurons within the same column and a long-range excitation with neurons in other columns. This suggests that signal receiving neurons in a column are in competition with each other (see for example, Mountcastle, 1975; Szentágothai, 1977; Eccles, 1981; Kohonen, 1989).

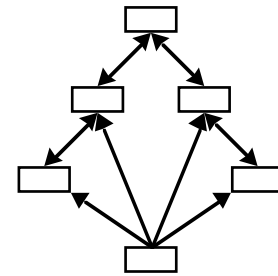


Figure 1. Sample multi-modular network illustrating recurrence and replication of substructures.

The suggestion is then that incorporating modularity into a connectionist model might provide a coarse initial architecture on which learning imparts a finer structure (Murre, 1992). Further advantages are an increased stability of learned representations and a reduced interference by subsequent learning, due to reduced plasticity of the network architecture (French, 1991; Grossberg, 1982, 1987; McCloskey and Cohen, 1989). In addition, modularity provides a form of task decomposition, because information processing is distributed over the modules and not only over the weights of the network. Task decomposition allows for solving a complex computational task by dividing it up into simpler subtasks and then combining their individual solutions (Haykin, 1994).

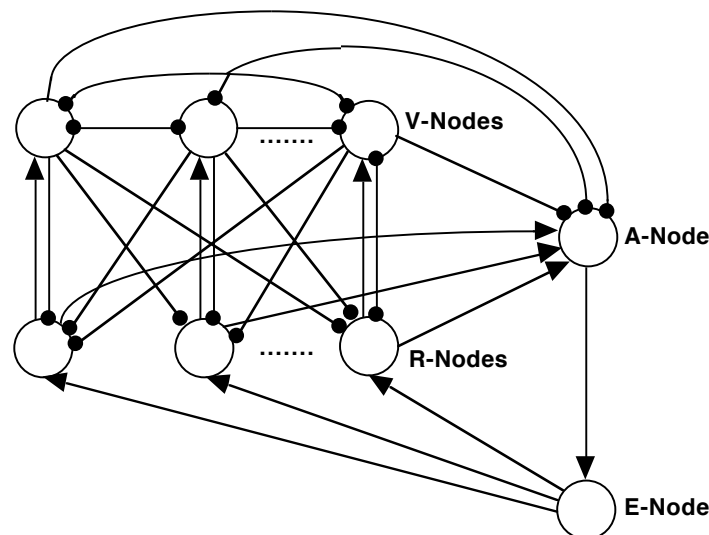


Figure 2 Schematic drawing of the internal wiring of a CALM module. Shown are the three internal node categories.

Modularity has therefore been made an important ingredient in Murre, Phaf, and Wolters' (1992) neural network architecture. A CALM network is built up with several inter-connected modules, although single-module networks are also a possibility (Figure 1; see also section 5). While the weights on the links between modules are modifiable, i.e. learning proceeds by adapting these weights, the modules themselves are fixed in structure and function. CALM relies on a several other important principles. The first one is that "categorization and learning are controlled by a *noise*-driven search process, that is structured through *competitive* selection." (Murre et al., 1992; Murre, 1992). Another principle is the *locality* principle which

says that all processing should be governed only by locally available information, to avoid the problem of central control. There also is the *arousal* principle which covers the idea that the arousal level in a CALM module is determined by the relative novelty of an input pattern. The latter principle implements the distinction between elaboration and activation learning, or a novelty dependent learning rate, by signaling when a new pattern is presented. The former type of learning results in the formation of new associations while the latter type merely strengthens existing associations (Graf and Mandler, 1984).

## 2. Architecture of CALM

A Categorization And Learning Module consists of four mutually exclusive functional groups of nodes as shown in figure 2. Information enters a module via representation nodes (R-nodes). In other words, connections between modules are actually connections between the R-nodes from different modules. The other nodes in CALM implement a competitive mechanism such that there is always one winner for each input pattern.

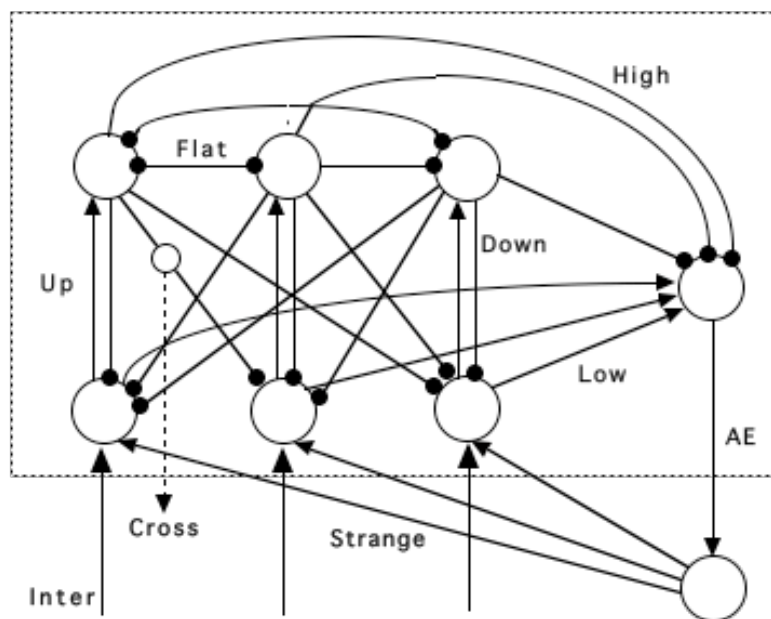


Figure 3 Schematic drawing of the wiring pattern.

Veto- or V-nodes have an inhibiting function. Every V-node receives excitatory activation from exactly one paired R-node and will in turn inhibit all other nodes inside a module to trigger competition. Because some inputs may be learned before and some inputs may be new to the net, another mechanism is needed to make it possible to distinguish between learning new information and maintaining old information. The amount of activation of the arousal- or A-node indicates the level of competition in a module, because it receives activation from all R-nodes. A previously learned input already has a winning R-node, whereas new input is likely to activate several R-nodes at once. The level of activation of the A-node in turn determines activation of the external- or E-node. The E-node is important in two ways. First, it sends random activation pulses to all R-nodes based on the amount of activation it receives. These random pulses force the competition mechanism to decide on a winning R-node in the case some R-nodes are equally activated (especially at the start of learning). This constitutes a noise-driven search process for new representations in CALM. In a similar fashion, Kam-Chuen, Giles, and Horne (1996) have shown that applying a controlled amount of noise during training of dynamically driven recurrent nets may improve convergence and generalization performance. Second, E-node activation modifies the learning parameter in the module, such that learning is low when arousal is low and high when arousal is high. In other words, if the

pattern presented has been encountered before weight modification is moderate, but if the pattern is new, weight adaptation should be more dramatic (cf. elaboration vs. activation learning).

The activation flow inside a module depends on the values of the fixed weights on the connections between the groups of intra-modular nodes (figure 3). Several types of fixed weights can be distinguished based on which nodes are connected. The values of these weights are quite flexible, provided they maintain a competition mechanism. For example, both the cross and down weights connect V-nodes to R-nodes, but in the latter a paired R-node is inhibited. Therefore, if a paired R-node wants to win competition cross weights should be much more inhibiting than the down weights (see Appendix I for a listing of parameter values). The activation of each node is calculated according to equation (1) given below and is a function of the activation of the node at a previous time step or iteration and the new incoming excitation. Because activation update in the net is discrete,  $(t+1)-t$  is the unit time interval or iteration time step. Three components can be identified in the activation rules. The first one,  $(1-k)a_i(t)$  denotes the autonomous *decay* of the activation of a node. If there is no incoming excitation, the activation of a node will decrease to zero with rate  $k$ . The second component,  $e_i/(1 \pm e_i)$  is a sigmoidal function which squashes the input excitation to a value between 0.0 and 1.0. Finally, the last sub formula makes sure that either the minimum or maximum activation is reached asymptotically.

$$\begin{aligned}
 a_i(t+1) &= (1-k)a_i(t) + \frac{e_i}{1+e_i} [1 - (1-k)a_i(t)] \text{ for } e_i \geq 0 \\
 a_i(t+1) &= (1-k)a_i(t) + \frac{e_i}{1-e_i} [(1-k)a_i(t)] \text{ for } e_i < 0 \\
 \text{in which } e_i &= \sum_j w_{ij} a_j(t)
 \end{aligned} \tag{1}$$

Learning in a CALM network consists of modifying the weights between modules. Connections between two modules are full, which means that every R-node in one module is connected to every R-node in the other. Weight update proceeds iteratively according to variation of Grossberg's learning rule (Grossberg, 1976), which is in itself an extension of the basic Hebbian learning rule. The basic principle of the Grossberg learning rule is that when calculating the new weight, it takes into account the total excitation of the receiving node. This Weber-law rule makes it possible to discriminate between different inputs that may even map non-orthogonal inputs into separate categories (Carpenter and Grossberg, 1986). The learning rules in CALM is then:

$$\Delta w_{ij}(t+1) = \left[ \sum_f a_f \left[ \sum_j K_{max} w_{ij}(t) a_j - L(w_{ij}(t) - K_{min}) \right] \right] w_{ij}(t) a_f \tag{2}$$

In this,  $a_f$ ,  $a_i$ , and  $a_j$  stand for  $a_f(t)$ ,  $a_i(t)$ ,  $a_j(t)$ , respectively;  $w_{ij}(t)$  is the interweight between R-nodes  $j$  and  $i$  (from  $j$  to  $i$ , both in different modules),  $w_{if}(t)$  is the interweight from a 'neighbouring' R-node  $f$  (of  $j$ ) to R-node  $i$ ,  $\Delta w_{ij}(t+1)$  is the change in weight from  $j$  to  $i$  at time  $t+1$ . Note that  $f$ ,  $i$ , and  $j$  must be R-nodes.  $L$  and  $K_{max}$  are positive constants.  $K_{max}$  determines the maximum value of an interweight, which may be approached asymptotically by  $w_{ij}$ .  $K_{min}$  indicates the minimum value of the interweights. Murre et al. (1992) only use positive weights, but Gibbons (1995) argues that negative weights should be allowed to prevent convergence for patterns that are counterexamples of learned categories (see Appendix II).

The first term within the large parentheses is always positive and represents increases in weight. The second term is responsible for all decreases in the weight. A connection from an inactive node ( $a_j = 0$ ) to an activated node ( $a_i$ ) will always decrease since the second term will

be non-zero. The increase/decrease ratio can be controlled by the  $K_{max}/L$  ratio. Furthermore,  $\mu_t$  represents the Hebb parameter, controlling the learning rate in the module, and is equal to

$$\mu_t = d + w_{\mu E} a_E \quad (3)$$

where  $d$  is a constant with a small value determining base rate learning,  $w_{\mu E}$  is the virtual weight from E to  $\mu_t$  (from the E-node to the learning parameter), and  $a_E$  is the activation of the E-node. Note, that this dependence on arousal implements the distinction between elaboration and activation learning (Graf and Mandler, 1984). However, the learning rate increases linearly with the activation of the E-node, which might not always be suitable, especially when at the start of the competition all R-nodes are active. To prevent this fast increase to maximum weight value, a Gaussian function is introduced instead. The idea behind this is that weight modification is low when arousal is either at a minimum or at a maximum and high in between. In other words, only learn when learning is necessary:

$$\mu_t = d + w_{\mu E} \cdot 0.0 \frac{(a_E - 0.5)^2}{0.25} \quad (4)$$

Because an interweight should be confined to the interval  $[K_{min}, K_{max}]$ , a complete description of the learning rule must include the following condition:

$$w_{ij}(t+1) = \max(\min[w_{ij}(t) + \mu_t w_{ij}(t), K_{max}], K_{min}) \quad (5)$$

The important properties of the learning rule are that high-weight values at time  $t$  tend to limit the increases in weight, and that  $\mu_t$  is made dependent on the activation of the E-node. Another feature of the learning rule is the influence of weighted background activation in the adjustment of the weights (the second term within large parentheses in the learning rule). In case of high background excitation (many nodes other than  $i$  feeding activation to node  $j$ ) increases in weight between  $i$  and  $j$  will become smaller and weights may even decrease. The effect is an adaptive downscaling of all weights, whenever the total input to the node will be too high. This may happen, for instance, if many modules are connected to one module, or if high input activations are present. The downscaling provides a mechanism that prohibits an overall increase in weights to some saturation level which eventually may paralyze the entire system (Murre, 1992; Murre et al., 1992).

Adaptation of weight is also related to the initial value of all learning weights in the net. In CALM weights are not randomized at the start of learning, but instead initialized to some fixed value. Murre (1992) has proven that after prolonged learning the weights will reach an equilibrium value. Because this equilibrium weight is only dependent on the parameters  $K_{max}$ ,  $L$  (assuming  $K_{min} = 0$ ), and the number of active inputs, we can use Murre's formula to derive the initial weight value for each module:

$$w_0 = \frac{\sqrt{1 + 4K_{max}Ln} - 1}{2Ln} \quad (6)$$

in which  $n$  is the number of neighbouring nodes ( $n \geq 1$ ). The underlying principle is that when the number of R-nodes is large, the initial weight is best kept low to prevent large changes in weights that might paralyze the net.

The values of internal weights together with the values for all other parameters are shown in Appendix I. These values were used as a standard in all experiments. Compared to the parameters originally described in Murre et al. (1992), the following has changed. First, the down weight has been changed from -1.2 to -0.2 to allow winning R-nodes to become highly active, and as such have a stronger propagation effect on connected modules. Secondly, the ER weight was decreased from 0.5 to 0.1. The ER weight connects the E-node to the R-

nodes and provides a random excitation to the R-nodes that serves as a tie breaker in dead-locked competition. Unfortunately, this can cause random convergences when a large random excitation occurs during a low activation portion of the update process. This reduced ER-value significantly increases the stability of R-node representations.

In addition, a slower learning rate is used, since it was found that the original values from Murre et al. (1992) not only cause CALM to show preferences to patterns that appeared early in the training cycle, but could also shift R-node representations over extended learning periods and, as such, harm the stability of learned representations. The simulations in this work use a value of 0.000001. We also reduced the virtual weight between the E-node and the learning rate to 0.0005, which makes learning more balanced as weight-changes are lower. Other CALM parameters are the same as in Murre et al. (1992), i.e. the decay of activation  $k$  is set to 0.05, the minimum and maximum value of learning weights is 0.0 and 1.0, respectively, and learning competition  $L$  is set to 1.0.

### 3. Functioning of CALM

A CALM module is characterized by three functional processes (Figure 4). First, we have the excitatory process triggered by stimulation of the R-nodes. Activation of this part will in turn trigger both the arousal and inhibitory parts of the module. V-nodes inhibit each other and are therefore responsible for competition inside the module. They also inhibit unpaired R-nodes, reducing the activity of these sources of competing V-nodes. The arousal process in turn makes resolution of competition possible and it also determines the learning rate. Three processes can be distinguished that play a role in the working of CALM (see figure 3). These processes are the excitatory, inhibitory and arousal processes. The first process is the activation of the R-nodes and is triggered by the stimulation of inputs to the module. These stimulations come from either other modules, the E-node or from receptor-nodes. The main function of the excitation process is the activation of the two other systems, namely the inhibitory system and the arousal system.

The excitatory system triggers the activation of the V-nodes, which constitutes the inhibitory process. An activated V-node will inhibit other V-nodes, the R-nodes and the A-node. Because of the mutual inhibition between V-nodes there will be competition between them. There is also a recurrent lateral inhibition between a V-node and the non-matching R-nodes, contributing to the competition by reducing the activity of these activation sources of the competitor V-nodes. The arousal process is triggered by the activation of R-nodes and reduced by the subsequent activation of V-nodes. It has two important functions. The first is to make resolution of competition possible. That is, when several R-nodes receive the same amount of activation the competition cannot be decided, but by the addition of random activations to these nodes the amounts of excitation will no longer be the same and competition will result into one winner. Moreover, the general purpose of deliberately introducing noise is to create stable noise resistant representations. The second function is to control the learning rate. The learning rate is proportional to the E-node activation. If much learning is required, as is the case with new unlearned patterns, then the weights change quickly for the E-node will be very active. In the case of already learned patterns, learning rate is low, just sufficient for strengthening of the representation, because then only one or few R-nodes are active.

The single most important feature of a CALM module is its ability to categorize input activation patterns autonomously (Murre et al., 1992). The learning of a CALM module is an instantiation of unsupervised learning, i.e. the

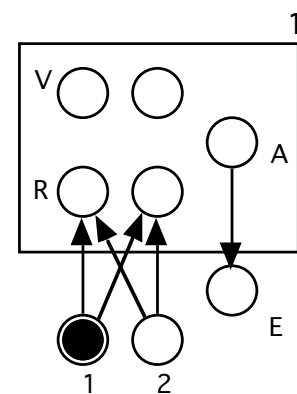


Figure 5: a CALM network with one module. The module consists of two R-nodes and the pattern is presented to the nodes 1 and 2. In the figure node 1 is activated.

network is not told what its output should look like. Categorization in CALM is operationalized as the association of a certain input pattern with a unique R-node that is then said to represent the pattern. During and following the categorization process learning takes place, which preserves the association between pattern and R-node by adjusting the interweights to the R-node (ibid.).

In figure 6 this process is visualized. This example illustrates the working of CALM. Each subfigure is a snapshot of a simple CALM network, as shown in figure 5, at iterations 1, 2, 3, 6, 11, 13, 17, and 28. The size of the dark rectangle in every node indicates its activation level. The input is delivered by two nodes 1 and 2 of which node 1 has activation 1 and node 2 is not activated. Figure 4 shows the network in its initial state: all nodes have zero activation and the interweights all have the same initial value. In the first iteration the excitatory process is activated by the stimulus. The arousal and the inhibitory processes are not yet activated. Note that both R-nodes have exactly the same activation value, because they receive the same amount of input due to the equal weights.

In the next iteration the inhibitory and arousal processes are activated. Both V-nodes are activated equally. The equal activity blocks solving the competition (there is no winner), but this is resolved because in the following iterations the E-node is also activated and noise or random activations on the connections to the R-nodes are added. By this chance process the first R-node receives more activation than the first one. We also see that after the sixth iteration the first V-node is more activated than the second V-node, and consequently the second R-V-pair loses activation. In the next iterations the network converges to a stable activation state where the first R-node represents the input pattern.

Note that during the whole process the connections of input node 2 to the module lose weight, because the learning rule makes weights of connections having no activation decrease. Furthermore, this example also shows the development of increased weights on connections between activated nodes. In the first 13 iterations both R-nodes are activated and remain in oscillation until the E-node noise has resolved the deadlock situation. After this the connections to the first R-node will strengthen.

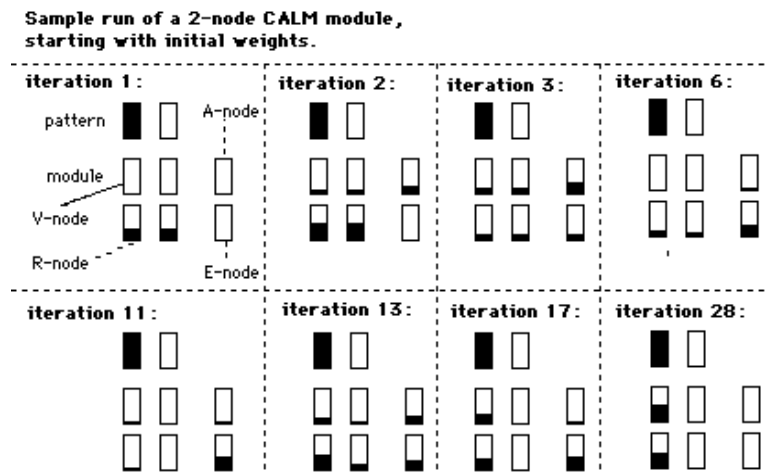


Figure 6. Flow of activation in a single module at the start of a simulation.

In the last few iterations, when the activation of the E-node decays and the representation of the pattern by the first R-node is established, elaboration learning is replaced by activation learning in which representations are only strengthened. It will be clear that in subsequent presentations of the same input pattern the first R-node has a direct advantage over the second because the weights to the first R-node are larger than the weights on the connections to the first one. So, because there is little activation of the E- and A-nodes, learning rate will be lower and further changes in weights will be much smaller (this is activation learning). In this way, repeated presentation of a pattern leads to the same categorization, and this

categorization will be reached much faster. The weight matrix that results from the first presentation is as follows (recall that all weights are initially equal, namely 0.50. Entry 1.1 denotes the weight of the link of the first pattern node to the first R-node):

from\to	1	2
1	0.656661	0.542989
2	0.402359	0.477211

It can be observed that the weights from the first pattern node to the first R-node are increased to the value 0.656661. The weights to the second R-node are also, but in a lesser extent, increased. This is the result of the learning rule which includes background activation in the calculation of the new weights. Note that the weight from the second pattern node to the first R-node is decreased stronger than the weight to the second R-node.

In figure 6 it is shown what happens if we present another pattern,  $\langle 0, 1 \rangle$ . Because the weight from the second pattern node to the second R-node is stronger than the one to the first R-node, as a result of the first presentation, the second R-node thus immediately has advantage above the first. Convergence is then achieved much faster as the second R-node already has more activation. Here the E-node only needs to strengthen the activation of the winning R-node. Because the other weights (from the first pattern node to the second R-node and from the second pattern node to the first R-node) do not transfer activation their values will decrease at every presentation of the patterns, until it reaches the minimum value 0. The following values show the change in weights after increasing presentations of patterns (for 50 iterations each):

after presentation of pattern $\langle 0, 1 \rangle$	from\to	1	2
	1	0.649805	0.455564
	2	0.417520	0.620888
after presenting both patterns for 2 <sup>nd</sup> time	from\to	1	2
	1	0.787323	0.323580
	2	0.295712	0.772979
after 10 presentations of both patterns	from\to	1	2
	1	0.990922	0.025970
	2	0.018500	0.987245

#### 4. Supervised learning

Supervised learning with CALM networks is possible by providing teaching signals to the module designated as an “output” module. During unsupervised learning, the activities of R-nodes are modulated by the randomized activity of the E-node. In the module that has to learn with supervision, the modulation from the E-node is cut off and replaced by a teaching signal. For example, if R-node  $i$  has to represent the current input, the value of its activation is incremented by  $w_{ER} \cdot 1.0$ , in which  $w_{ER}$  is the weight from E-node to R-node. All other R-nodes receive the negative modulation  $w_{ER} \cdot -1.0$  to suppress possible stronger competition. The advantage of using modulated pulses instead of plainly setting the activation of the desired winning R-node to 1.0 is that competition is still kept alive such that weight changes are relatively stronger. In addition, the learning rate for connections to the feedback module may be slightly increased. For recurrent connections from the feedback module, the background activation in weight update is turned off in order to have the feedback information become a stronger determinant for learning new associations.

Consider a CALM network with two modules of sizes 4 and 2, respectively, and an input module of size 8. We present binary patterns that encode line lengths as in the following table:

input patterns								uns.		sup.	
								A	B	A	B
1	0	0	0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	0	1	0	1	0
1	1	1	0	0	0	0	0	1	3	0	3
1	1	1	1	0	0	0	0	0	3	1	2
1	1	1	1	1	0	0	0	0	3	0	3
1	1	1	1	1	1	0	0	0	2	1	2
1	1	1	1	1	1	1	0	0	2	0	1
1	1	1	1	1	1	1	1	0	1	1	2

The network was presented 5 times with the whole input set, with each pattern iterated for 50 times. Without teaching signals, the network, in one sample run, arrived at the clustering shown in the table under the heading “uns.” If we provide teaching signals that separate odd length from even length patterns, then, starting from the weight configuration after the previous training, the network acquires the clustering shown under the header “sup.”, also after 5 epochs. Figures 7 and 8 show how the weights have altered under supervision.

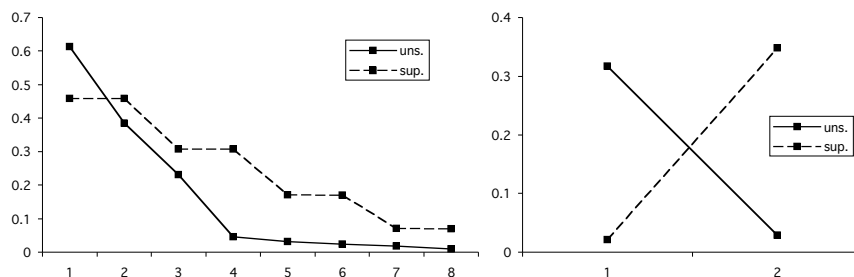


Figure 7. Left hand panel shows the weights from the input module to an R-node in the output module, before and after supervision. Right hand panel shows the weight from an R-node in the 4-node module to an R-node in the output module, before and after supervision.

Other methods of supervision are possible. Tijsseling (1998) provides a special feedback module that sends correction signals to a connected module when the winner of that module does not agree with the teaching pattern. Alternatively, Berthouze & Pic (2001) add an R-V pair to a module and force it to be the winner by setting its activation to 1.0. The problem with this latter approach is that adding a new R-V pair disturbs the modules weights, causing previously taught representations to become unstable. With multi-modular networks, this shift in weights will also disrupt representations in connected modules. Moreover, adding forced R-V pairs as winners causes the competition mechanism of the module to become paralyzed.

### 5. Multi-modular architectures

The real power of a CALM module is in its co-operation with other modules. The design of a multi-modular networks (such as in figure 1) is based on several principles (Happel and Murre, 1994). First, learning and categorization might improve as the induced categories are more compatible with the category structure of the task domain. For example, a coarse categorization in a small module can interactively facilitate a more fine-grained categorization in a larger module. A comprehensive way of looking at multi-modular nets is in terms of circuits of R-nodes. With the presentation of each input pattern the network converges on a set of activated R-nodes that are connected by learning connections. Learning therefore creates circuits of associations between active R-nodes from all modules (cf. Hebb’ cell assemblies (1949). In addition, the presence of various size modules in a net helps to hierarchically categorize inputs. The advantage is that with the presentation of an input, evidence for a coarse category structure present in a small module helps to rule out irrelevant categories in a

larger module by activating the associated R-nodes in this module. Similarly, if subsequent convergences on two or more R-nodes in a large module coincides with convergence on the same R-node in a small module, categories of the large module becomes associated with the single common category in the small module.

Performance of a multi-modular net is improved further by replicating substructures. For example, in figure 1, the left hand substructure of the net is exactly the same as the right hand substructure. The rationale behind this is that categorization in a subsystem, compatible with the induced category structure, proceeds faster than incompatible categorizations in other subsystems, because optimal categorization is characterized by quicker convergences on winning R-nodes as it causes less competition in modules (Happel and Murre, 1994). As a consequence, a compatible categorization will be a greater determinant of the overall categorization in the whole net. This influence of compatible categorization can be enhanced by installing recurrent connections to the network, such that fast, optimal categorization in one pathway may correct slower (and initially suboptimal) categorizations in other pathways. Recurrence can also introduce top-down and bottom-up information streams in a network, which can interactively affect categorization at all levels. Bottom-up connections of a relatively fast pathway may create an “expectancy” in the top module, which in turn might disambiguate local information processing via top-down connections (McClelland and Rumelhart, 1981). Happel and Murre (1994) prove that the probability of correct categorization in a network increases with the addition of each replicated structure.

The above design-principles for building networks consisting of more than one CALM modules are derived from neuroscience, psychology, and computational learning theory. The way the brain is organized restricts the set of functions it can perform. Only because of its highly organized architecture the brain is able to perform a myriad of functions and yet to maintain a compact size. The structure of the brain has evolved to be able to capture as many regularities of the world around us as possible (Happel & Murre, 1994). The first principal form of global architectural organization of the brain that comes to mind is its layered structure. Subsequent layers of neurons (not to be confused with neural layers in the neocortex), arranged in a hierarchical fashion, form increasingly complex representations. Such multi-stage processing of information is present for example in the visual system, this consists of several layers from the retina towards the cortical centers V1 up to V5 (Hubel & Wiesel, 1959, 1962, 1965; Zeki, 1992). The second architectural organization is the presence of multiple parallel processing streams or pathways (Livingstone & Hubel, 1988; Kosslyn, Flynn, Amsterdam, & Wang, 1990; Zeki, 1992). In investigating the processing of *what* (the identity of objects) and *where* (the location of objects) in the natural visual system, Ruckl, Cave, and Kosslyn (1989) showed with a series of simulations that separate processing pathways improve learning, because overlap and interference of representations are decreased. This separate processing is also present in the brain. Distinct streams of information processing in the brain thus allows for the independent processing of different aspects of information both between and within sensory modalities.

When magnifying a part of the brain a highly regular structure is found in the cortical mini-columns (Mountcastle, 1975). These neural modules consist of about a hundred cells with a particular organization of connections, namely few excitatory between module connections and many mainly inhibitory within module connections. The cortical mini-column has been proposed as the basic functional modular unit of the cerebral cortex (Mountcastle, 1978; Szentágothai, 1977; Eccles, 1981). Not much is known about the exact working of the cortex, but Happel and Murre refer to Creutzfeldt (1977) who suggests that it functions as a cooperative network in which all areas are subject to the same general structural principles. The specific function of any cortical area is largely defined by the origin of their afferents and the destination of their efferent connections (Happel & Murre, 1994).

The modular architecture of the brain is also present at the functional level as has become evident from a wide range of psychological studies. For example, the anatomical division of

the brain into the major hemispheres is paralleled at the functional level by hemispheric specialization. Each hemisphere is devoted to partly different mental functions. Within each hemisphere specific functions are organized into anatomically separate regions (Brodmann, 1909). Kandel and Schwartz (1992) have shown that even the most complex functions of the brain can be localized at least to some degree. The modular division of function is probably advantageous because of the minimization of mutual interference between simultaneous processing of different information and execution of different tasks. Different tasks are executed by distinct streams of modules which do not interfere with each other. Similar tasks, however, access the same stream and are therefore subject to interference (Allport, 80).

## **6. Nonlinear dynamics of multi-modular architectures**

Although a single CALM module works as a linear categorizer, chaotic behavior (May, 1976; Ruelle, 1990) occurs in hierarchical multi-modular structures, which have also been termed fractal neural architectures (Heemskerk et al., 1992). The functions performed by these recurrent, interactive networks produce non-linear input/output mappings that would require an number of hyperplanes and consequently, an infinite number of nodes and layers in order to implement them in a feedforward network. The behavior of multimodular CALM networks is chaotic in nature and that this chaos may serve several useful purposes such as creating fractal category structures and avoiding sequential interference (Happel & Murre, 1995; Tijsseling, 1998).

The connection structure of a CALM module provides an intricate, unstable mechanism in which R-V pairs form circuits of coupled oscillators. In general, coupled oscillators are known to generate a variety of dynamical regimes in which periodic, quasi-periodic, and chaotic dynamics designate universal classes of behavior. Oscillatory phenomena are prevalent, they have been shown to occur in the visual system (Gray & Singer, 1989) and in the olfactory system (Skarda & Freeman, 1987). It has been shown that the phase behavior of oscillatory neural circuits can account for the dynamic binding of visual and auditory stimulus features as well as for figure-ground separation in networks.

The presence of coupled oscillations in a connectionist system is one of a set of features which have been argued by Skarda & Freeman (1987) to be required in order for chaotic behavior to emerge in such systems. The other features are local feedback and local inhibitory connections. Although these are not present in ART networks (Skarda & Freeman, 1987), they do exist in CALM networks. Naturally, the presence of chaotic behavior should be beneficial for the system. Many authors have argued for several reasons why chaos serves the performance of a dynamic system. Tsuda (1992) gives a list of the functions chaos may have:

- i. novelty filter (Skarda & Freeman, 1987; Tsuda, 1992; Happel & Murre, 1995)
- ii. explorative deterministic noise (Skarda & Freeman, 1987; Yao & Freeman, 1990)
- iii. memory searcher (Tsuda 1992; Koerner, Tsuda, & Shimizu, 1987)
- iv. dynamic store-room of LTM (Tsuda, 1991)
- v. catalyst for learning (Skarda & Freeman, 1987)
- vi. non-linear pattern classifier (Freeman, Yao, & Burke, 1988; Happel & Murre, 1995)
- vii. STM generator (Nicolis & Tsuda, 1985)

During learning, the network develops nonlinear category structures as a consequence of multiple information streams flowing through the various modules. The emergence of these so called fractal attractor basins can be attributed to oscillations and the presence of multiple attracting regimes which are present in the dynamics of the system. This kind of learning mechanism reduces sequential interference from mapping overlapping input patterns in separate output categories. Moreover, an advantage of the emergence of fractal category boundaries is that they can, in principle, make infinitely more distinctions between input patterns than can a linear categorization, just because fractals are the only mathematical objects that can specify an infinite surface within a finite space (Happel & Murre, 1995).

Fractal boundaries often emerge during initial phases of learning. Happel & Murre show that for a simple two-module network trained on patterns  $[0,1]$  and  $[1,0]$  the category boundaries get smoother with more presentations of the same input set and will eventually become a diagonal line through the two-dimensional space. As the network does not know the optimal category structure it has to maintain hold of the many varying input values. Chaos thus seems to play a role in the categorization of stimuli and also in the reduction of sequential interference. Chaos also appears to be responsible for the way representations are stored. The relatively slow adaptation of primary category boundaries may selectively facilitate the long term storage of ambiguous input patterns, which lie close to a category boundary (Happel & Murre, 1995). This may yield a highly adaptive short term storage in combination with stable long term storage of information, an organization that is also a characteristic of memory in the brain (Tsuda, 1991; Nicolis & Tsuda, 1985). Chaotic properties of the nonlinear dynamics of recurrent networks can also serve as exploratory deterministic noise or a memory searcher. Input patterns which have not been trained before, very often (depending on how the similarity is with the stored patterns) enter chaotic orbits. The network, in this state, enables itself to explore the existing weight space in search of best matching representations.

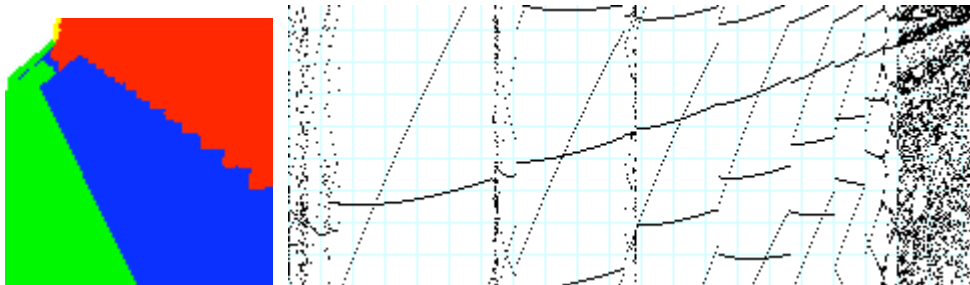


Figure 8. Left hand panel shows a convergence map, indicating by color for each  $x$  and  $y$  in the input pattern, the final winning R-node of a specified module. The right hand panel shows the change in sum of total activation over all R-nodes when one dimension of the input pattern is changed gradually. Each column shows the total activation for 25000 iterations, ignoring initial transients.

Several methods exist to determine the nonlinear dynamic properties of the network. Tijsseling (1998) provides a Lyapunov exponent calculation for the CALM algorithm, that indicates whether the network's response to a given input is chaotic, periodic or fixed point. Happel & Murre (1995) plot the index of the winning node for any  $x$  and  $y$  in the input, which reveals properties of category boundaries (Figure 8, left hand side). By varying one dimension of a given input pattern and plotting the total activation over all R-nodes, bifurcation diagrams can be created (Figure 8, right hand side). These are useful in showing sudden transitions in network dynamics.

## 7. Self-organization with CALMMaps

So far, we have discussed plain CALM modules that are competitive only in nature. It is, however, also possible to have CALM modules self-organize their representations, such as to create a topological map of the input space (Phaf, Den Dulk, Tijsseling, & Lebert, 2001). Implementing self-organization in CALM provides solutions to some problems that may occur in CALM modules. For example, separation of patterns in CALM is a complicated function of the distance between and overlap of different patterns as well as the size of the module and the dimensionality of the patterns. Particularly in larger modules and with larger patterns, even a large Euclidean distance between patterns may not be sufficient for a stable distinct categorization. This property of CALM may be particularly damaging when modular networks are exposed to ecologically plausible stimuli, because such stimuli will tend to contain information on many different attribute dimensions (e.g. Phaf, van der Heijden, & Hudson, 1991) of which only a few may differ sufficiently to discriminate stimulus objects. In fact, in a network that segregates these attribute dimensions along different pathways of

successive categorizations, the initial distance between stimulus objects may be insufficient to obtain separation with the CALM-procedure (Phaf, Den Dulk, Tijsseling & Lebert, 2002).

A second problem is the randomness of the search of new representations in CALM. If a new pattern differs sufficiently from the already represented patterns a new representation is selected at random. Because the separation criterion more or less acts as an absolute threshold, after the initial comparison of all patterns, further presentation of the dataset will not increase separation substantially in CALM. If we had a relative separation criterion that also depends on the mutual relations between the patterns, categorization could go on until all patterns are evenly distributed over a module (Phaf, Den Dulk, Tijsseling & Lebert, 2002). CALM Map is a solution to these problems.

CALM Map enforces a topological structure on the representations by using the notion of *neighbourhood.*, i.e. the distance between nodes becomes another determinant of learning. In CALM Map the highest activation is not selected with Euclidean distance measure but by competition due to the actual lateral inhibition between units with activations determined by the weighted sum rule. The observed advantage of CALM Map is improved categorization because the stretching property enables a continuous separation. This differs from both CALM and Kohonen maps. CALM will eventually commit itself to a once obtained categorization and a Kohonen map will not continue to stretch after full separation is reached (Phaf, Den Dulk, Tijsseling & Lebert, 2002). CALM Maps also show a total absence of retroactive interference when the interfering set of patterns forms an interpolated set of the initial data set (ibid.).

An additional advantage of CALMMaps is when the number of R-nodes exceeds the number of input patterns, representations will be separated maximally such that committed (i.e. actually representing an input pattern) and uncommitted R-nodes alternate. The uncommitted nodes then interpolate between the representations of the neighbouring nodes. According to Ritter (1993), this interpolation behavior has the advantage of learning from very few examples. To encode intermediary patterns only the extremes of the full range would, in principle, need to be presented.

The topological mechanism is implemented by replacing the Cross and Down weights in the module with a inhibitory gradient of weights determined by the distance between a V- and R-node. The function is a Gaussian according to a theorem by Erwin et al. (1991), which states that when a part of a Gaussian function is used as convex inhibition gradient and the “full width at half height” of the Gaussian equals the number of units, convergence of the self-organizing network will be optimal.

$$w_{ij} = ne^{-\frac{\sigma^2 \cdot (i-j)^2}{n}} - n \cdot 1 - c \quad (7)$$

where  $w_{ij}$  denotes the inhibitory weight between the  $i$ th R-node and the  $j$ th V-node,  $i - j$  is the distance between the two nodes,  $n$  is the number of R-nodes, and  $c$  is the default inhibition value (equal to Down weight). The remaining parameter values are the same as in the standard CALM (See appendix I). The value of  $\sigma$  determines the steepness of the Gaussian curve. Figure 9 shows a sample Gaussian function for  $\sigma = 0.2$ ,  $c = 0.2$  and module size 9.

The topological ordering in CALM Maps can be of two types: a line and a ring topology. The first type orders the representations along

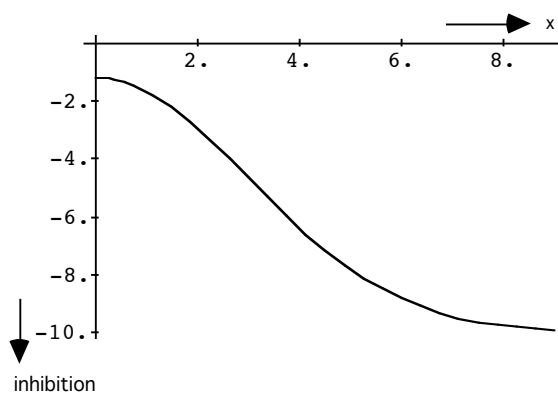


Figure 9:  $\sigma$  is 0.2;  $c = 0.2$ ;  $0 \leq x \leq 9$ .

a line starting with the first node of the module end ending with the last. The latter type treats the first and the last nodes of a module as neighbours, i.e. there is no beginning or end in the ordering. To illustrate the ordering process in a module with a ring topology a small experiment was conducted. A set of nine patterns as shown in table 1 were presented in permuted order for 25 epochs to a module of size 9, each pattern iterated 50 times. Between the presentations all activations were set to zero. After 24 presentations the representations were ordered on a one-dimensional scale, the results are shown in figure 10.

1:	111100000000
2:	011110000000
3:	001111000000
4:	000111100000
5:	000011110000
6:	000001111000
7:	000000111100
8:	000000011110
9:	000000001111

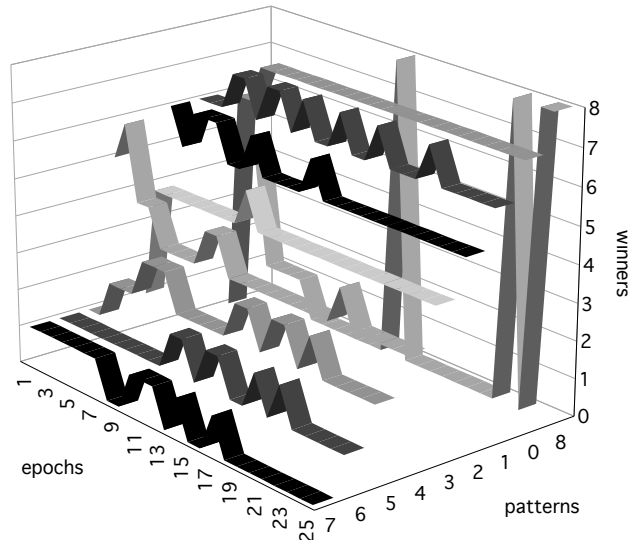


Table 1. Patterns

Figure 10. Learned ordering in CALMMap for patterns in Table 1. Shown are the winning R-nodes after each epoch.

It may be interesting to consider the ordering process. Due to the graded inhibition the competition process first yields a broad shallow activity pattern around the most excited R-V-pair (here node 2), because this R-node initially receives the least inhibition with equal activations of all R-nodes (which is the case due to the initially equal learning weights). Increasing modular weights cause the R-nodes closest to the winning node to gain larger inputs, resulting in higher excitation of V-nodes and even stronger inhibition of the other nodes. Thus, the “activity bubble” automatically decreases during all phases of pattern presentations. At the same time, however, the winning R-node with the minimum total inhibition drifts towards the boundaries and each node will become specifically tuned to a particular input pattern. The "activity-bubble" and the corresponding change in connection weights will tend to push the dip in the total inhibition (as a function of R-node position) sideways. This mechanism, which is enhanced by the random activation from the E-node, causes the representations to spread maximally over the nodes in the module (ibid.). In Kohonen maps such a maximal separation process is not present.

This ordering process also shows that the distinction between elaboration and activation learning in CALM Maps can also be seen to conform nicely to Kohonen’s account of the computational requirements for topological self-organization (Kohonen, 1988). In this account two subsequent phases in the development of a map are mentioned. Initially, all patterns are represented on one (central) node, which is not the case with the standard CALM module. In the first phase the patterns are roughly split up along the R-nodes. This constitutes the ordering phase. After this initial order is established, the network gradually reaches a maximum spacing of representations. This phase is called the convergence phase by Kohonen. This can be seen to correspond with the two forms of learning in CALM. The ordering phase corresponds to elaboration learning and the convergence phase to activation learning.

We performed a series of experiments to investigate the functioning of CALM Map and to compare it with CALM (Phaf, Den Dulk, Tijsseling & Lebert, 2002) These experiments will not be described here, I will refer to the original articles for a detailed description. Only the results will be presented here. In general the CALM Map has a better performance than the CALM module. The effects of the size of the module and the overlap and the Euclidean distance between patterns were investigated. Concerning the size of the module the results for both networks did not differ substantially except for the interpolation effects with module sizes larger than the number of patterns as mentioned elsewhere in this chapter. Overlap between patterns hardly affected categorization in CALM Map in contrast to CALM. Finally, for small (Euclidian) distances between patterns CALM Map separated representations better than CALM. Because topological mappings frequently occur in the nervous system (Gilbert & Wiesel, 1989, Amari, 1980) we have an argument for using CALM Map in modeling lower level cognitive capacities such as perceptual categorization, because this or a similar stretching process presents an optimal and continuous differentiating ability for unsupervised learning in real neural systems.

## 8. Processing sequential information

Because CALM networks allow for recurrent connections in a modular structure, it has the inherent capacity to learn sequential patterns. Gibbons (1995) provides a simple extension to allow for a temporal-to-spatial transformation in the same way as time-delay networks. This consists in allowing for a time-delay connection between modules such that state information of a past time can be maintained. Normal connections between modules never store state information, because competition requires that all module activations to be reset before a new input pattern is presented. If activations are not reset, competition is non-existent due to modules already having converged.

By introducing time-delay connections, we allow the activations of R-nodes after convergence to be stored and accessible to a connected module. A module that receives a time-delay connection updates its activations and weights based on the connected module's activations from the previous convergence. Equations (1) and (2) are accordingly updated. In equation (1), the incoming weighted activation for time-delay connections becomes:

$$e_i = \sum_j w_{ij} a_j(t-k) \quad (8)$$

in which  $k$  is the time since last convergence. And from equation (2) we derive the following weight update rule for time-delay connections:

$$\Delta w_{ij}(t+1) = \sum_i a_i(t) \left[ \sum_j K_{max} \Delta w_{ij}(t) \right] a_j(t-k) \left[ L(w_{ij}(t) - K_{min}) \sum_{f \neq j} w_{if}(t) a_f(t-k) \right] \quad (9)$$

For time-delay connections two different time-scales are implemented using the above equations. The short time scale handles the intra-modular activation update and the long time scale handles the update of weights based on incoming activations at time  $t-k$ . This means, that in effect, inputs are not propagated to a next module until the current module has converged. With normal connections, activation flows through modules with every update cycle.

Normal and time-delay connections can then be combined into a sequence learning modular network, an illustrative example of which is shown in Figure 11. Input consists of sentences fed into the network letter by letter. The first letter of the sentence is presented to a module cluster "11", which is a multi-modular substructure with normal intermodular connections to ensure correct categorization of inputs. The output of these module clusters is propagated to the next cluster, "12", via a time-delay connection. This in turn links to "13".

The outputs of all three clusters, which will be encoding three subsequent letters in the input stream, is propagated using normal connections to another, higher level cluster “w”. This module cluster will then categorize inputs that span multiple time units, and would basically learn to encode the common, repeating patterns such as words or word segments. This hierarchical processing of input streams can be repeated with additional module clusters. Gibbons (1995) provides a simulation that replicates Elman (1990) recurrent neural network simulations of word acquisition. Gibbons was able to produce similar results using only single modules instead of module clusters, achieving 90% correct classification of words (Gibbons, 1995).

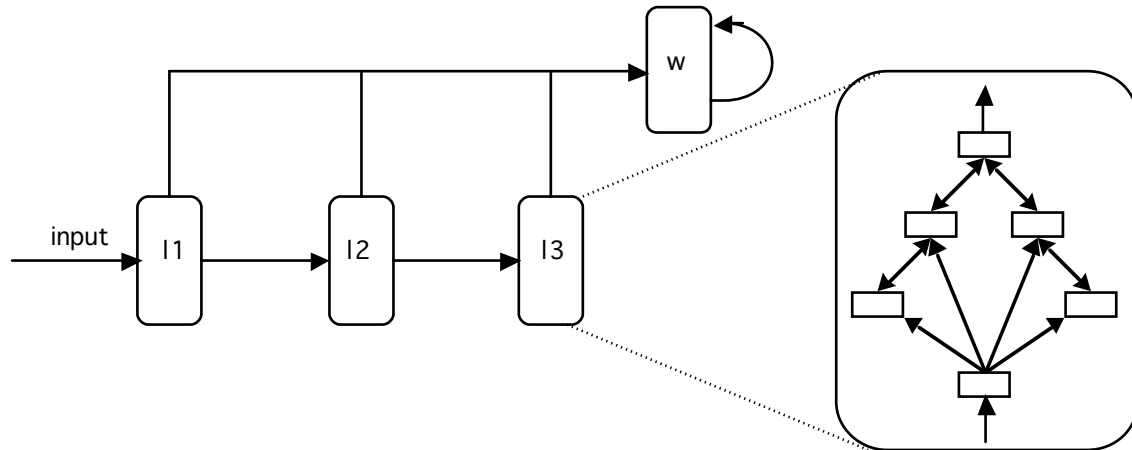


Figure 11: Illustrative time-delay CALM neural network (only partly shown). Input is a stream of letters forming a sentence. Hierarchical processing ensures categorization of letters, words, and sentences.

A problem with this approach to learning sequential information is that there are still only two time-scales or time-constants. With normal connections, the network processes input instantaneously and with time-delay connections it will process the information from the previous convergence. But what if we want to capture the varieties in timing information in the input stream? To achieve this, we modified the time-delay connections to also have a particular time-constant. This constant is an integer larger than 0 and indicates the delay between input presentations. Assume that a sequence  $p_1 \square p_2 \square p_3 \text{---} p_4 \square p_5$  is presented at corresponding times  $t, t + 1, t + 2, t + 3, t + 4, t + 5$ , then a delay connection with time-constant 2 will only be processing the inputs  $p_2$  and  $p_4$ . A set of time-delay connections with various time-constants will become highly useful for extracting various timing information from the sequence of inputs.

## 9. Autonomous addition and pruning of nodes

If the structure of given problem domain is known to the experimenter as well as the number of input samples, then the sizes of modules in a CALM network can be relatively straightforwardly set. However, not only do real world data usually have a structure that is yet unknown to the experimenter, the data samples can become quite large. For this reason, it would be beneficial to the network if it is able to adapt its module sizes to the current problem domain as its structure is being unfolded during learning. For example, if new samples are encountered that do not fit the existing category structure (such as the example given in Appendix II), the network could add more R-V pairs to encode for this new information. Additionally, when R-V pairs become obsolete over time, it is desirable to have them pruned to save up resources and increase computation speed.

This adaptive change in node numbers can be related to neurobiological neuron-growth and –pruning, in particular during development. For example, Rakic et al. (1986) found an increase in synaptic density across several regions of the Rhesus monkey cerebral cortex in early age (peaking at 2 and 4 months), which was followed by a sharp decline in later months. The

growth rate of neurons may be related to input activity according to a study by Lund et al. (1990). This is a relevant observation as it would allow activity-dependent mechanisms to partly regulate the neuronal structure. Quartz & Sejnowski (1997) sum several studies providing evidence for this suggestion.

Not only is the addition and removal of nodes in a neural network conformant with respect to neurobiological evidence, it is also relevant from a computational viewpoint. A demonstration by White (1990) shows how allowing a network to grow as it learns will give it the advantage of being able to essentially learn any arbitrary mapping. In fact, the addition of nodes solves the issue that the solution of weights in a fixed neural network architectures is a NP-complete problem (Blum & Rivest, 1988). Baum (1989) observed that structure-adding networks are complete representations, i.e. these networks can learn any problem in polynomial time. In the context of CALM, these considerations are only partly applicable, since we limit ourselves to only grow or shrink individual modules. In other words, we are not adding new modules to a given network, making the architecture semi-fixed.

Growing and pruning of R-nodes is implemented as follows. Since CALM drives on changes in attention and competition, each module already has an implicit activity-dependent mechanism available. The pruning of obsolete R-V pairs, for example, can be easily implemented by verifying an R-nodes activity level over time. We define the potential  $p_i$  for each R-unit  $i$  which has an initial value of 1.0 and which changes over time according to:

$$p_i(t) = \frac{p_i(t-1) \cdot t + a_i(t) \cdot a_E(t)}{t+1} \quad (10)$$

Equation (10) captures the activation of an R-node over time, giving an accurate estimation of its role in the network's overall performance. This equation causes the potential of an R-unit to slowly decay over time, while allowing sudden new activations to recharge the unit's potential. The activation from the E-node plays a crucial role in the R-nodes potential: If the E-node activation is high, competition is ongoing and nodes cannot be pruned. Only when the E-node activation settles to baseline can the potential of an R-node significantly decrease. When the potential of an R-node reaches below a pre-determined threshold (e.g. 0.001), the R-V node pair is pruned.

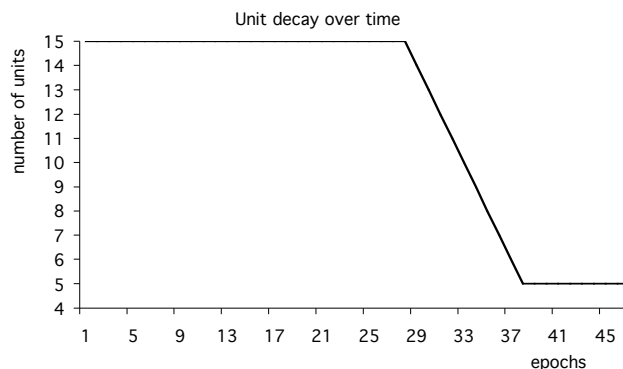


Figure 12: Decrease of module size during learning in a CALM module with an initial size of 15 R-V pairs. After acquiring the structure of the stimulus domain after about 25 epochs, unused R-V pairs are pruned until the module reaches a size of 5.

The pruning method was applied to the problem domain of Appendix II. The pattern set contains of 10 patterns, designed in such a way that each subsequent three patterns form a distinct cluster, but the last pattern matches each of these three clusters and hence should be assigned to a different R-node. We use a single module CALM network with the size set to 15. Figure 12 shows the decrease in module size over epochs for a single run. The network initially maintains the same size, but after about 25 epochs, when a suboptimal category structure has been acquired, uncommitted R-node potential decay rapidly, causing a quick

drop in module size until only 5 R-V pairs are left. Over several different runs, the same results are observed and category structure consistently place the last pattern in a separate category and clustering the other 9 in three separate categories, occasionally using an extra R-node.

A more difficult issue is to determine when a module needs additional R-V pairs to address a problem domain of which the underlying structure is not compatible with the network's acquired knowledge. An example is the addition of the 10th pattern in the example of Appendix II to a module of size 3. Because this 10<sup>th</sup> pattern does not fit any of the 3 categories formed by the other 9 patterns, it would have to be assigned to a new R-node. To have the module autonomously detect the necessity to supply additional resources, the most logical way would be to access the level of competition in the module, which is part of Equation 10. Because competition levels may vary strongly over one epoch, the calculation of the new potential in Equation 10 implements a variation of a moving average. The potential of an R-node is determined by its history and the effect of the current competition level.

We are interested in the potential of R-nodes that represent input patterns that are incompatible, such as pattern 10 from the example of Appendix II and any of the other 9 patterns. Such R-nodes have a potential level that is relatively high compared to the other nodes. In fact, when for a given module, the maximum potential is a predefined percentage higher than the next largest voltage ( $(p_1 \square p_2/p_1) \cdot 100.0 > T$ ), an R-V pair is added. For example, if we train a module of size 3 on the pattern set of Appendix II, the voltages for a given run are 0.133, 0.132, and 0.170. The latter value corresponds to the R-node that represents one category for 3 input patterns as well as the incompatible 10<sup>th</sup> pattern. With  $T$  set to 10.0, this means that the module size will increase to 4.

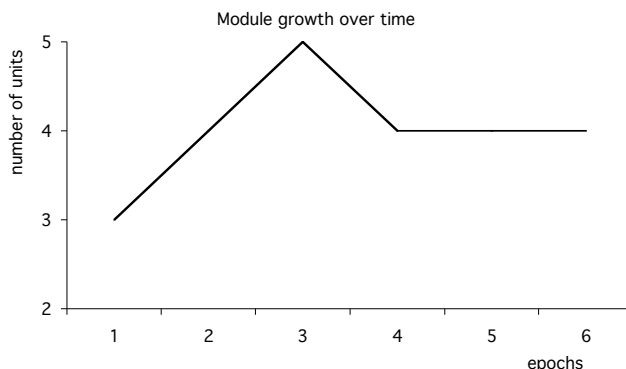


Figure 13: Growth and shrinkage of a module with an initial size of 3 R-V pairs. An optimal module size was quickly reached after only 4 epochs.

Figure 13 illustrates a sample simulation for the above module, in which growing and pruning occur after each epoch. The module size was initially set to 3. After presentation of the complete pattern set for the first time, the module size increases to a size of 5 in two epochs, but reduces again to a size of 4, which is then maintained until the end of training. The optimal category structure was acquired for this module size. In fact, adding the pruning and growing mechanisms makes the use of negative weights obsolete.

## Bibliography

- Allport, D.A. (1980) Patterns and actions. In: G.L. Claxton (Ed.), *New Directions in Cognitive Psychology*. London: Routledge & Kegan Paul.
- Amari, S.-I. (1980) Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42: 339-364.
- Baum, E.B. (1989) A proposal for more powerful learning algorithms. *Neural Computation* 1: 201-207.
- Blum, A. & Rivest, R.L. (1988) Training a 3-node neural network is NP-complete. In: *Advances in neural information processing systems*, ed. D.S. Touretzky. Morgan Kaufmann.
- Berthouze, L. & Pic, M. (2001) Emergence of language in interactive systems, *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, Tucson (USA).
- Brodmann, K. (1909) *Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues*. Barth, Leipzig.
- Carpenter, G.A., & Grossberg, S. (1988) The ART of adaptive pattern recognition by self-organizing neural networks. *Computer*, 21: 77-88.
- Creutzfeldt, O.D. (1977) Generality of the functional structure of the neocortex, *Naturwissenschaften* 64: 507-517.
- Damasio, A.R., Damasio, H., & Van Hoesen, G.W. (1982) Prosopagnosia: Anatomic basis and behavioral mechanisms. *Neurology*, 32: 331-41.
- Eccles, J.C. (1981) The modular operation of the cerebral neocortex considered as the material basis of mental events. *Neuroscience*, 6, 1839-1855.
- Elman, J.L. (1990) Finding structure in time, *Cognitive Science*, 14, 179-211.
- Elman, J.L. (1993) Learning and Development in Neural Networks: The Importance of Starting Small. *Cognition* 48, 71-99.
- Erwin, E., Obermayer, K., & Schulten, K. (1992). Self-organizing maps: Stationary states, metastability and convergence rate. *Biological Cybernetics*, 67, 47-55.
- Freeman, W.J., Yao, Y., & Burke, B. (1988) Central pattern generating and recognizing in olfactory bulb: a correlation learning rule. *Neural Networks*, 1: 277-278.
- French, R.M. (1991) Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks. Indiana University, Bloomington. CRCC Technical Report 51.
- Gibbons, T.E. (1995) *Unsupervised categorization of sequential data*. Thesis for the degree of Doctor of Philosophy, Department of Computer Science and Operations Research, North Dakota State University.
- Gilbert, C.D. & Wiesel, T.N. (1989) Columnar specificity of intrinsic horizontal and corticocortical connections in cat visual cortex. *The Journal of Neuroscience*, 9: 2432-2442.
- Graf, P. & Mandler, G. (1984) Activation makes words more accessible, but not necessary more retrievable. *Journal of Verbal Learning and Behavior*, 23: 553-568.
- Gray, C.M. & Singer, W. (1989) Stimulus-specific neuronal oscillations in orientation columns of the cat visual cortex. *Proceedings of the National Academy of Science, USA*, 86: 1689-1702.
- Grossberg, S. (1976) Adaptive pattern classification and universal recoding, II: Feedback, expectation, olfaction, and illusions. *Biological Cybernetics*, 23: 187-202.
- Grossberg, S. (1982) *Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control*, Boston, MA: Reidel Press.
- Grossberg, S. (1987) Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11: 23-63.
- Hanson, S.J. & Burr, D.J. (1990) What connectionist models learn: Learning and Representation in Connectionist Networks, *Behavioral and Brain Sciences* 13, 471-518.
- Happel, B.L.M. & Murre, J.M.J. (1994) Design and evolution of modular neural network architectures. *Neural Networks*, Vol. 7, No 6/7, 985-1004.
- Happel, B.L.M. & Murre, J.M.J. (1995) Evolving complex dynamics in modular interactive neural networks. *unpublished*.
- Haykin, S. (1994) *Neural Networks: A Comprehensive Foundation*. New York: Macmillan College Publishing.
- Hebb, D.O. (1949) *The Organization of Behavior*, New York: Wiley.
- Heemskerk, J.N.H., Murre, J.M.J., Melissant, A., Pelgrom, M., & Hudson, P.T.W. (1992) MindShape: A neurocomputer concept based on the fractal architecture. In: I. Aleksander & J. Taylor (Eds.), *Artificial Neural Networks II: Proceedings of the ICANN-92*, Brighton, UK. Amsterdam: Elsevier: 1483-1486.
- Hubel, D.H. & Wiesel, T.N. (1959) Receptive fields of single neurons in the cat striate cortex. *Journal of Physiology*, 148: 574-591.
- Hubel, D.H. & Wiesel, T.N. (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160: 106-154.
- Hubel, D.H. & Wiesel, T.N. (1965) Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, 28: 229-289.
- Kam-Chuen, J., Giles, C.L., & Horne, B.G. (1996) An analysis of noise in recurrent neural networks: Convergence and generalization. *IEEE Transactions on Neural Networks*, 7, 6, 1424-1438.
- Kandel, A. & Schwartz, (1985) J.H. *Principles of Neural Science*. NY: Elsevier.

- Koerner, E., Tsuda, I., & Shimizu, H. (1987) Parallel in sequence - Toward the architecture of an elementary cortical processor. In Albrecht & Jung & Mehlforn (Eds.), *Mathematical Research: Parallel Algorithms and Architectures*: 37-47. Akademie-Verlag: Berlin.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.
- Kohonen, T. (1988). *Self-organization and associative memory* (2nd ed.). Berlin: Springer Verlag.
- Kohonen, T. (1993). Physiological interpretation of the self-organizing map algorithm. *Neural Networks*, 6, 895-905.
- Kohonen, T. (1995). *Self-organizing maps*. Berlin: Springer Verlag.
- Kosslyn, S.M., Flynn, R.A., Amsterdam, J.B., & Wang, G. (1990) Components of high-level vision: A cognitive neuroscience analysis and accounts of neurological syndromes. *Cognition*, 34: 203-277.
- Livingstone, M. & Hubel, D.H. (1988) Segregation of form, color, movement, and depth: Anatomy, physiology, and perception. *Science*, 240: 740-9.
- Lund, J.S., Holbach, S.M., & Chung, W.W. (1990) Postnatal development of thalamic recipient neurons in the monkey striate cortex: II. Influence of afferent driving on spine acquisition and dendritic growth of layer 4C spiny stellate neurons. *Journal of Comparative Neurology* 309: 129-140.
- Luria, A.R. (1973) *The working brain: an introduction to neuropsychology*. New York: Basic.
- May, R.M. (1976) Simple mathematic models with very complicated dynamics. *Nature* 261: 459-67.
- McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, 88, 375-407.
- McClelland, J.L., Rumelhart, D.E. & the PDP Research Group (1986) *Parallel Distributed Processing*. Vol 1 & 2, Cambridge, MA: MIT Press.
- McClelland, J.L., McNaughton, B.L., & O'Reilly, R.C. (1994) *Why there are Complementary learning systems in the hippocampus and neocortex: Insights from successes and failures of connectionist models of learning and memory*, Carnegie Mellon University & The University of Arizona, Technical Report PDP.CNS.94.1, March 1994
- McCloskey, M., & Cohen, N.J. (1989) Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24: 109-165
- Mountcastle, V.B. (1978) An organizing principle for cerebral function: The unit module and the distributed system. In: G.M. Edelman & V.B. Mountcastle (Eds.), *The Mindful Brain*. Cambridge, MA: MIT Press.
- Murre, Jacob M. J. (1992) *Learning and Categorization in Modular Neural Networks*, Harvester Wheatsheaf, Hempstead, UK.
- Murre J.M.J., Phaf, R.H., & Wolters, G. (1992) CALM: Categorizing and Learning Module, *Neural Networks*, Vol 5, 55-82.
- Nicolis J.S. & Tsuda, I. (1985) Chaotic dynamics of information processing: the "Magic number seven plus-minus two" revisited. *Bulletin of Mathematical Biology* 47: 343-65.
- Phaf, R.H., Van Der Heijden, A.H.C., & Hudson, P.T.W. (1990) SLAM: A connectionist model for attention in visual selection tasks. *Cognitive Psychology*, 22, 273-341.
- Phaf, R.H, Den Dulk, P., Tijsseling, A.G. & Lebert, E. (2001) Novelty-dependent learning and topological mapping. *Connection Science*, 13(4), 293-321.
- Quartz, S.R. & Sejnowski, T.J. (1997) The neural basis of cognitive development: A constructivist manifesto. *Behavioral and Brain Sciences* 20, 537-596.
- Rakic, P., Bourgeois, J.P., Eckenhoff, M.F., Zecevic, N., & Goldman-Rakic, P.S. (1986) Concurrent overproduction of synapses in diverse regions of the primate cerebral cortex. *Science* 232:232-235.
- Ritter, H. (1993) Parameterized self-organizing maps. *Proceedings of the International Conference on Artificial Neural Networks*, Amsterdam, The Netherlands.
- Rueckl, J.G., Cave, K.R., & Kosslyn, S.M. (1989). Why are 'what' and 'where' processed by separate cortical visual systems? A computational investigation. *Journal of Cognitive Neuroscience*, 1, 171-186.
- Ruelle, D. (1980) Strange attractors. *The Mathematical Intelligencer*, 2, 126-37
- Rumelhart, D.E. (1989) The architecture of mind: A connectionist approach, In I. Posner (Ed.), *Foundations of cognitive science*, Cambridge, MA: MIT Press.
- Shallice, T., McLeod, P., & Lewis, K. (1985) Isolating cognitive modules with the dual-task paradigm: Are speech perception and production separate processes? *The Quarterly Journal of Experimental Psychology*, 37A: 507-32.
- Skarda, C.A. & Freeman, W.J. (1987) How brains make chaos to make sense of the world. *Behavioral and Brain Sciences*, 10, 161-195.
- Szentágothai, J. (1975) The "module-concept" in the cerebral cortex architecture. *Brain Research*, 95, 475-496.
- Tijsseling, A.G. *Chaos, self-organization & multi-modular architectures (Book review of Murre on categorization)*, psycoloquy.95.6.08.categorization.13.tijsseling, ISSN 1055-0143, in PSYCOLOQUY, Friday, 7 April 1995.
- Tijsseling, A.G. (1998) Connectionist models of categorization: A dynamical view of cognition. Ph.D. Thesis, Southampton University, Southampton, England.
- Tsuda, I. (1991) Chaotic itinerancy as a dynamical basis of hermeneutics in brain and mind. *World Futures*, 32: 167-184.
- Tsuda, I. (1992) Dynamic link of memory - Chaotic memory map in nonequilibrium neural networks. *Neural Networks* 5: 313-26.

- White, H. (1990) Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural Networks* 3:535-549.
- Yao, Y. & Freeman, W.J. (1990) Model of biological pattern recognition with spatially chaotic dynamics. *Neural Networks*, Vol. 3: 153-170.
- Zeki, S. (1992) The visual image in mind and brain. *Scientific American*, September 1992

## Appendix I: Parameter values

Up-weight	0.5
Down-weight	-0.2
Cross-weight	-10.0
Flat-weight	-1.0
High-weight	-0.6
Low-weight	0.4
AE-weight	1.0
ER-weight - used to be 'strange'	0.1
Virtual weight from e to mu	0.005
Base rate learning $d$	0.0005
Decay parameter activation rule $k$	0.05
Grossberg $K$ -parameter $K_{max}$ (max weight)	1.0
Grossberg $K$ -parameter $K_{min}$ (min weight)	0.0
Grossberg $L$ -parameter	1.0
Initial learning weight value	0.6
Auto decay of weights	0.0

## Appendix II: Learning with negative weights

Gibbons (1995) provides an illustration of the necessity of negative weights. Consider the following set of inputs and accompanying typical R-node representation:

111 000 000	1
110 000 000	1
101 000 000	1
000 111 000	2
000 110 000	2
000 101 000	2
000 000 111	3
000 000 110	3
000 000 101	3
100 100 100	?

Based on Euclidean distance, each subsequent three patterns will be represented together in a 3-node CALM module. The last pattern, however, is equally distance to any of the three groups and, therefore, would be represented on a different R-node. Since there are only 3 R-nodes, it will be randomly assigned a representation. By allowing, negative weights, this random categorization is prevented as the network will fail to converge. Below is a sample weight matrix which leads to a zero activation for the last pattern, with  $K_{min} = -1.0$ :

1	-0.126	-0.175	-1	-1	-1	-1	-1	-1
-1	-1	-1	1	-0.166	-0.174	-1	-1	-1
-1	-1	-1	-1	-1	-1	1	-0.119	-0.166