ONTOGENESIS IN NEURAL NETWORKS

Robert Pallbo

Lund University Cognitive Science,
Kungshuset, Lundagård,
S–223 50 Lund,
Sweden.
E-mail: robert.pallbo@fil.lu.se

Abstract. When working with a neural network, it is advantageous to let it extract as much information as possible from the environment. We also want the network to perform better at this task. One way of achieving this is to make the perceptions of the network dependent on its current knowledge. This means that the network should be able to find structures in structures or to make categories of categories. This paper is an attempt to accomplish these goals by crafting the units to be feature correlators rather than feature detectors. A central component in the proposed construction is the use of spontaneous (or background) activity of the units as an important influence on the process.

1 INTRODUCTION

The methodology of using neural networks to simulate various kinds of processes is nowadays recognized in several areas of research. One of these areas is biological modeling, which brings together cognitive scientists, neurophysiologists and biologically oriented AI researchers. One of the main research interests in this area is to construct models of complete agents, which are fully equipped with sensors, motor effectors and an appropriate set of behaviours. The present paper belongs to this category.

The aim of the study is twofold. First, it is an attempt to formulate a neural network theory for autonomous agents. Second, it is an attempt to increase our understanding of the mechanisms underlying biological networks. The approach taken, is related to the work of Skarda and Freeman (1987) in two respects. First, background, or spontaneous, activity is given a crucial role in the process. Second, the system does not learn in the traditional sense, but rather, as Werner expressed clearly in his comments (Werner, 1987): "[T]he neural structure *uses* information to *create* its own internal states, which acquire meaning." A consequence of this is that the structure in the network is not formed by the external stimuli, but *evolves* from inside. This evolving process is what I recognize as the ontogenesis of a neural network.

This perspective on a neural network introduces some constraints. First, as the goal is an autonomous system, the learning must be unsupervised and incremental. Furthermore, the system must work with a continous flow of input. Finally, we want to design a system that perceives and acts in the real world, or, at least, in the *same* world. That is, the task at hand is not to construct a system that constructs abstract categories from the input, but a system that perceives the states of an environment and subsequently – and continuously – acts in this environment.

I will first present some general principles that I think the system should be based upon. Second, I will discuss how spontaneous activity can be used in the system. Third, I will bring up implementational issues and, finally, the approach taken in this paper will be compared to some excisting networks.

2 A MINIMAL NEURAL LEARNING SYSTEM

In the spirit of Occam's razor, we would like to construct a system that is based on as few principles as possible. This will minimize the human preprogramming needed and introduce generality of the system. We do not want to design a full system for any task in which we use an autonomous agent; this can be obtained by traditional programs. Rather, we want to isolate general principles for a system that can be used in order to *evolve* the system. That is, we want to construct a system that, in a sense, is able to program itself!

For the design of a self-organizing, evolving, and minimal neural learning system I propose three basic principles to be required:

(1) Spatial chunking,

(2) Temporal chunking, and

(3) Learning modulation

Lund University Cognitive Studies – LUCS 24 1993. ISSN 1101–8453.

By (1) we reach the traditional achievements of neural networks. Principle (2) allows encapsulation (or categorization) of events and sequences, and, finally, (3) is necessary as a mean to direct the learning. These mechanisms should be operating all over the network and hence not isolated in different modules.

2.1 SPATIAL CHUNKING

In the field of connectionism, chunking of spatially distributed patterns, or, more generally, categorization is a big issue. Almost all networks are constructed to somehow categorize the input. The very output of such a network is usually a category formed by the input data. This is achieved by means of various techniques – may it be supervised or self-organized.

The importance of spatial chunking, from my point of view, is that it allows a more economic representation. This is especially important in a neural network since the representation is distributed. By using categories, it is possible to use a smaller number of units to represent a given situation. This means that more patterns can be differentiated by the system if the number of units is kept constant. Further, the risk that two or more simultaneous patterns should overlap and interfere with each other decreases. In other words, categorization allows more information to be comprehended by the network.

Another virtue of spatial chunking is that it allows generalization. When a particular pattern is perceived, it is categorized in relation to the previous learning that has taken place in the network. This process will erase some information carried by the pattern and introduce new information not previously present. This generalization has its price however. The network might be blind to the differences in two patterns that we (or itself) would like it to differentiate. The two patterns will be categorized in the same way and thus acted on as if it were the same situation. This problem illustrates the importance of (incremental) learning in the system. If a system is initially (perceptually but not sensory) blind to the differences between two patterns, we would like there to be a possibility for the network to learn the difference.

2.2 TEMPORAL CHUNKING

Given that the perception of the world does not consist of independent snapshots but of a continuous flow of patterns, we would like our system to handle such input. Such systems, however, are few in number even if the applications are numerous. Temporal problems are to a great extent neglected in theoretical connectionism.

As in the case of spatial patterns, we would like to chunk temporal patterns or sequences. This would also allow the representation to be more economical, and we would achieve the same benefits as in the spatial case. By categorizing *spatial* patterns we achieve a decreased usage of the medium – space. In the case of *temporal* patterns, we would like to reduce the usage of its medium – time. How should this be done? The traditional solution is to transfer the temporal pattern into a spatial one by means of shift registers, unit activity decay or time-delaying units. Those approaches, however, have to my knowledge no support from biology. Further, it introduces limitations on the length of the sequences that the system can handle. This ability becomes dependent on the number of such shift registers.

The best way to approach temporal chunking is, in my opinion, to represent *time* in the pattern flow by *time*. This is not as obvious as it seems since this means that temporal patterns cannot be handled by the same mechanisms as spatial patterns.

Temporal chunking is a delicate problem, and it should not be studied in isolation from the spatial chunking or in isolation from learning modulation. Temporal patterns incorporate both sequences of categories as well as transformations of patterns. The issue of how these should be implemented is not currently solved. The current research of the author is to a great extent focused on this specific issue.

2.3 LEARNING MODULATION

The third principle needed in a neural learning system concerns an ability to modulate, or direct, the learning. As with most neural networks, the adjustments take place at the synaptic level. All learning is manifested by changes in the parameters of the synapses. This approach works fine when the learning considered is local like the learning of categories from sensor input. Learning from behaviour, however, is a different case. When an agent is acting in its environment, it can get good, bad, or no feedback. It is desired to take these values into account for the adjustment of learning at the synaptic level. Therefore, we need a mechanism for communication between the higher levels of abstraction in which behaviour occurs and the lower synaptic level in which the actual learning takes place. The learning modulation is thus related to reinforcement learning. Notice however, that it is the system *per se* that evaluates the responses from the environment as good or bad. A particular environmental state might be good for one system but bad for another. Furthermore, a particular environmental state can be both good or bad for the same system on different occasions. It is possible to assume that the system can be in various motivational states. If the agent is hungry, certain behaviours are more appropriate than if it is not hungry. A model of motivation should therefore be an integrated part of the system (Balkenius, 1993).

3 AN EVOLVING NETWORK

An evolving network must have a basis from which to evolve. The task of the evolvement is to move this platform into more and more complex structures. In order to achieve this, there must (a) be a way for the network to search for new structures and (b) be a mechanism that can maintain the progression by moving the platform. The search is accomplished by neuronal spontaneous activity and in some respect by the network architecture. The mechanism to secure the progress is the long-term learning, modulated somehow by the value that the progress gives the system (agent).

I would now like to outline some principles of a network capable of evolving the complexity of the categories. The network is based on two fundamental issues, *spontaneous activity* (or noise) and, *correlation* of activity. Note that by noise I do not mean simulated annealing (Ackley et al., 1985), but a constant level of spontaneous activity. It might seem strange to *add* noise to a system, since noise is usually regarded as something unwanted and bad. But noise actually has a central function in the network is proposed here. If noise is not added, the network will enter a state of silence and no activity at all will take place.

Please notice that the system as a whole has not yet been subjected to any computer simulations. A network using spontaneous activity has been simulated though and will be described briefly in the section below.

3.1 SPONTANEOUS ACTIVITY

Most connectionist networks (if not all) are driven by their input. Without an external activation of the sensors, the network remains silent. This is not the case, however, in biological networks. In a situation where no obvious external input is present, a recording from, e.g., the visual cortex will reveal that the area is displaying what is called spontaneous activity (Squatrito et al., 1990; Sato et al., 1989). By using spontaneous activity as the driving force, the system becomes independent of external activation. The process is of course still highly affected by external stimuli.

In Pallbo (1992; 1993), spontaneous activity was used in a model of visual motion detection and line extraction. This model employs no learning, but it illustrates how the usage of spontaneous activity can reduce the complexity of the computation in the individual nodes as well as simplify the architecture of the network. In the model, the nodes are cooperating in order to carry out the detection. Indeed, no node is able to perform detection alone. The idea of the model is to allow the nodes to "peek" at their neighbours and rely on that information for their task. The nodes detect motion at one direction only. This allows the nodes to "peek" at those neighbours that will be signalling the same motion some short time before themselves. A node will signal that motion has occurred if it gets this information from those neighbours *and* if some activity is present in the small field of the visual scene to which the node is connected. This model of motion detection has been successfully tested in computer simulations with video recordings as inputs.

The virtue of using spontaneous activity in that model is that it allows the system to detect what is stable in the picture. Usually, motion detection involves the search for what has *changed* in the picture. The consequence is that in such approaches noise is a big problem. The reason for this is that noise, in addition to moving objects, is constantly changing from snapshot to snapshot. If the search is made for stable patterns propagating through the picture, noise is no longer a problem. The problem is instead to find new patterns in motion, which is the task of the spontaneous activity.

3.2 CORRELATION

Suppose that a spontaneous activation of a node happens to be correlated with an external event. Then, the active nodes at that point could be regarded as indicating this event. The synapses that contributed to the activation of the node will strengthen their possibility to mediate the same information again (Fig. 1). If the same nodes are activated again in correlation with the same external event, then the strengths will grow even stronger. Of course, all synapses that are contributing to the activation at one occasion are not significant as indicators for the event. These synapses, however, will therefore reduce their participation in this categorization when the target node repeatedly gets activated without the involvement of these synapses.

The nodes in the network should not be viewed as feature *detectors*, but feature – or event – *correlators*. The importance of their firing lies in that it is correlated with some external or internal event (cf. Neural Darwinism, Edelman, 1987). In this manner, the nodes' activity encapsulate the more distributed information that indicates an event, and the representation becomes more economical. An alternative view is that by modifying the connections in the network, the system reprograms itself into feature detectors. This self-programming is occurring over and over again as the network evolves. Correct correlations that appear spontaneously are maintained by means of the synaptic plasticity while incorrect correlation is not.

4 IMPLEMENTATIONAL ISSUES

In the previous section it was suggested that correlation and spontaneous activity should be the main features of the network. In this section, I will try to go into more details on how this could be implemented.

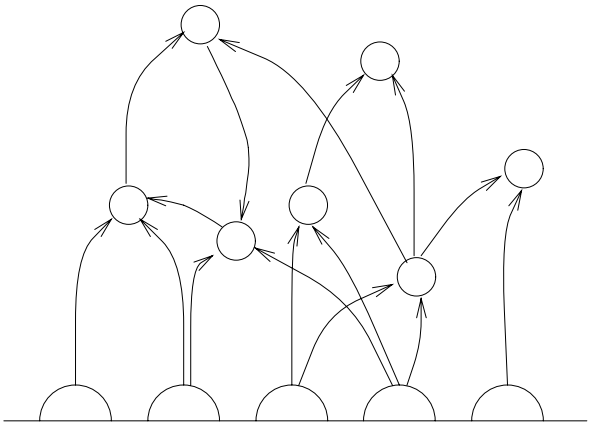

Figure 1: A network with sensors (half-circles), nodes (full-circles) and connections (arrows). The connections are established (strengthened) after that correlation has been detected. The figure illustrates both nodes connected directly to the sensors as well as nodes more distant to the raw input data.

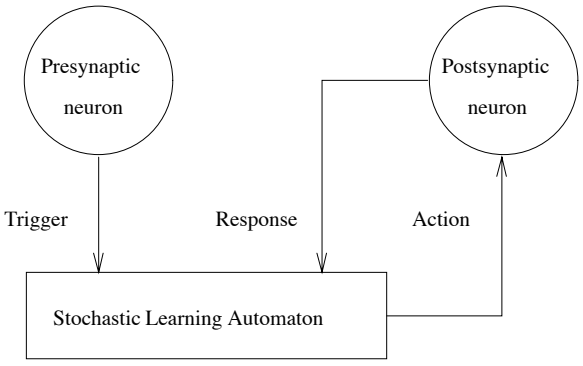

Figure 2: The model of a synapse as a stochastic learning automaton.

4.1 THE SYNAPTIC LEVEL

The learning is taking place at the synaptic level. A single synapse might, or might not, contribute to a unit's activation. It is not sufficient that just one synapse contributes to the activation of a unit. Many synapses must cooperate in order to cause an activation. Therefore this task can be viewed as a multi-agent cooperative learning problem, and hence, we can employ methods from the field of game theory.

Let us take the perspective of a single synapse. Its environment consists of the pre- and postsynaptic nodes. The synapse will know whenever the presynaptic node is activated, and the synapse can act by either mediating this signal to the postsynaptic node, or remain silent. The probability of the behaviour that the synapse will perform depends on the previous learning at the synapse. A useful model is stochastic learning automata (Tsetlin, 1973; Narendra and Thathachar, 1974; Barto, 1989) illustrated in Figure 2.

The presynaptic node acts as a trigger to activate the synapse. When activated, it must decide whether to stimulate the postsynaptic node or not. This is determined by a stochastic variable $P(m) \in [0, 1]$, where $P(m) = p(action = 1 \mid trigger = 1)$ and m is a variable affected by learning. After the synapse has been triggered and either transmitted a signal to the postsynaptic node or remained silent, it can read the state of that node through a response signal. With this information, the automaton can either strengthen the possibility to do what it did, decrease it, or just leave the probability unchanged. What change to make is dependent on both the learning strategy and whether the synapse is excitatory or inhibitory. It is not within the scope of this paper, however, to go into the details of these variations. Rather, it is the task of practical experiments to find out what learning rule to use.

An extension to the learning taking place in the synapse is to split m into two parameters (cf. Balkenius, 1992). We could have $m = m_s + m_l$, where m_s is changed on a short term basis and m_l on a long term basis. m_s would increase more than m_l on every occasion, but also decrease faster. This way we achieve a short term memory and a long term memory.[1] The network would be highly affected by recent activity without forgetting learning that has taken place over a long time span.

4.2 THE NEURAL LEVEL

From the perspective of the synapse, the neuron was reduced to a more or less passive environment. What each neuron does is simply to add its input, and if the sum exceeds a critical amount, then the neuron fires, otherwise not.

Even if this behaviour is simple, we as observers can interpret the activity of the neurons as if they actually did compute some more sophisticated function (like motion detection). This is also how their activity will be interpreted by other neurons. Each neuron will look upon the others as holders of information. The neuron will in a sense search for those neurons that give relevant information for the computation done by the neuron, that is, neurons that are firing prior to itself, or, neurons that fire in *correlation* with it.

The neurons in the network would remain silent with this arrangement if it were not for the spontaneous activity. By this feature, the network becomes activated even if the activation in most cases is nonsense. But the network follows the rule that it is better to do

[1] The long and short term memory discussed here are not intended as a psychological model of memory.

anything than to do *nothing*; sometimes just anything will be right. This anything will later evolve to be more and more meaningful as the quality increases in the categories found. This is why the spontaneous activity is the *motor* of the network.

4.3 THE ARCHITECTURAL LEVEL

In the example of motion detection above, the architecture is crucial for the performance. If the excitatory lateral connections are not placed on only one side of the neurons, but on both, the neurons transform into line extractors. If these connections would be symmetric all-around the neuron, we would not get much of selectivity at all.

What we should learn from this case study is that not only the spontaneous activity and the learning are important for the evolvement of the network but also the architecture of the connections. Most probably, it is not possible to find a general architecture that will fit under all circumstances for all problems. This is especially true for those parts of the network that are closest to the sensors. At networks more distant from the sensors, we probably have to deal more with temporal encapsulation and a more general approach might be possible.

Another feature that should be present in the architecture is *variation*. Since we want different neurons, or groups of neurons, to evolve into different detectors (or information holders), this divergence must originate in variation of some kind. Variation of course, is also present in the input data and, not least, in the spontaneous activity. But for some computations to emerge, variation in the architecture is necessary (remember again the case of motion detection).

4.4 THE MODULAR LEVEL

In order to achieve a complete system, it is helpful to organize the network into modules of subnetworks. (cf. the areas of the cerebral cortex in mammals). We can have perceptual, effectual and associative modules. We can also have primary and secondary modules. The modules interact through (topological) projections (Fig. 3).

The arrangement of the network in modules allows the introduction of a mechanism that can constrain the communication between them, that is, a selectional or motivational system. This can either be implemented by letting this system affect the communication directly or by modulating the level of spontaneous activity inside the modules. By decreasing the level of spontaneous activity, the nodes in the module will be less able to ignite a detection. The details of such a system, however, are subjects of future research.

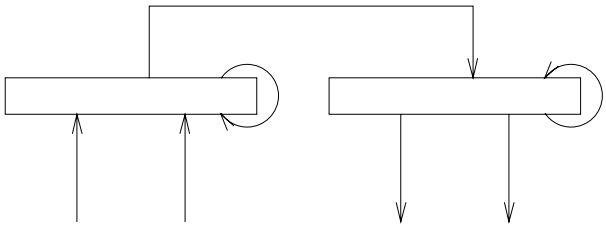

Figure 3: A simple modular network connected to perceptors and effectors. The perceptual module projects onto the effectual module.

5 A COMPARISON WITH TRADITIONAL NEURAL NETWORKS

Neural networks in general are well known for their ability to self-organize. For solving a specific task, the network is trained through learning by example. The same initial network can be trained for several distinct problems and the approach is thus very general.

The problem with neural networks is that the complexity that can be achieved in the categorization is very restricted. A single output, single hidden layer feedforward network is capable of learning any arbitrary mapping (White, 1990), and this generalizes straightforwardly to the multi-output multi-hidden layer case according to White. This is fine, but works only for supervised networks. For unsupervised networks, the situation is worse. Further, it is not a mere mapping from one domain to another that we strive for, but a system that can produce and organize long sequences of behaviour.

The general problem for unsupervised networks is the absence of a mechanism that will bring the self-organization beyond the simple categorization done on the raw input. It is desired that the network should be able to perceive new data in the light of its current knowledge, i.e., to evolve its complexity.

The limitations of most neural networks of today can be summarized in the following:

- The networks support communication and learning between layers of nodes, but not between modules. Hence, there is no guideline for how the network should be extended into more complex systems without an external observer interfering and supervising the internal representation of the system.

- The interaction between various scale levels are either absent or managed by an external algorithm targeted to a specific level. This has the consequence that higher levels that could emerge, do not have any possibility to affect the

other levels. Hence, such higher levels will not emerge.

- The learning algorithm is typically complex.

In the following sections, I will bring up a few related networks and point out their specific weak (or strong) points.

5.1 KOHONEN NETWORKS

A Kohonen network (Fig. 4), or self-organizing map, is a topologically ordered feedforward network in which all input nodes are connected to all output nodes (Kohonen, 1989). The weights are set to random values at initialization. When an input pattern is presented, one of the output nodes will be most activated. The connections to this node, and nodes in its immediate neighbourhood, get their strengths adjusted in order to give an even stronger response to the same pattern in subsequent trials. This is repeated for a large set of patterns, and if the size of the neighbourhoods and the weight adjustments are decreased during learning, the output nodes will become organized into clusters. Close nodes will therefore respond to similar patterns. The lateral interaction of the nodes in a neighbourhood can be realized under the assumption that later interaction between the cells are mainly inhibitory. In the computer simulations, however, this interaction is usually implemented externally to the network.

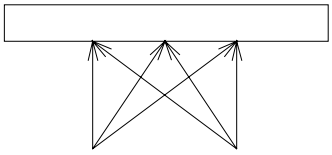

Figure 4: A one-dimensional Kohonen network with two-dimensional input. All input nodes are connected to every output node.

The architecture of Kohonen networks reveals that the categories formed in this system are made from raw data. Hence, the complexity of the categories (clusters) formed in this network is restricted. In order for the network to be useful, the input data is usually preprocessed in a suitable manner to bring forth the structures that one wants the network to detect. The network *can* be used in the application of complex neural networks, but just as a part of it. The theory of this network, however, gives no support when the maps are used as modules in a more complex system. The learning rule is restricted to one map and does not generalize itself in a straightforward manner to a modular version.

Moreover, Kohonen networks can be criticized for their external control of the parameter that decreases the size of the topological neighbourhood. This is not accurate if we desire biological realism. The division into a learning phase and a using phase lacks biological realism as well. This makes incremental learning impossible in this kind of network.

5.2 ART NETWORKS

An ART network is a sensory feature detector (Fig. 5). The network is able to learn new categories fast without forgetting previously learned categories (Carpenter and Grossberg, 1993). An ART network is basically organized into two layers, one input layer for representing sensory features and one output layer that represents categories. Principally, the network is a self-organizing feature map with the addition of feedback connections. These connections allow the activity in the network to reach a resonant state in which learning occurs.

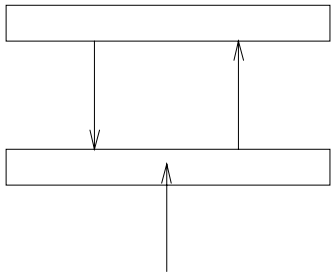

Figure 5: An ART network consists primarily of two layers plus some additional subsystems (not shown) used to deal with category mismatch. The formation of categories in the upper layer does not benefit much from earlier experience which is why the network does not evolve in its complexity.

There is no interaction within the output layer except for lateral inhibition. Hence, the process of forming new categories cannot benefit from those previously learned by the network. The categories formed in an ART network are invariable categories that are a direct result of the raw input data. The output nodes are, however, not totally independent. If a new input pattern is similar to a previously learned category, it will be categorized as this old category. If it is not sufficiently similar, a new category is formed.

An advantage of ART networks worth mentioning is its capability of incremental learning. Further, several ART3 networks can be connected in serial and/or in parallel. There is, however, no means for evolving the complexity in the network. Therefore, ART networks primarily lend themselves to adaptive neural machineries to which it is very suitable.

5.3 BOLTZMANN MACHINES

Some readers might object that a network along the direction of this paper is just a version of the Boltzmann machine network (Ackley et al., 1985; Hinton and Sejnowski, 1986). This is, however, incorrect. First of all, the aims of these two networks are different. When the Boltzmann machine is given an input, it strives for reaching a stable (final) state. When this state is reached, it is read off by a (to the network) external observer. The Boltzmann machine methodology makes use of noise in a manner very different from the approach taken in this paper. When the Boltzmann machine is given an input, it heads for the closest *local* minimum within the context of stable patterns. The designer of the net, however, would prefer the network to stabilize itself at the *global* minimum. Therefore, noise, or energy, is added to force the network out from the local minima. The hope is that by slowly decreasing the energy (simulated annealing) the network will reach the global minimum.

The approach taken in this paper does not primarily use noise as a mean to get out of local minima. There should be no such thing as a stable state in a biologically plausible network. Rather, noise is the originator of all detection occurring, that is, the activity patterns grow from noise by modulation from the sensory input. Further, the Boltzmann machine operates in modes, which is not the case here. A second problem is that the learning is not incremental. The two networks are thus rather different.

5.4 ELMAN NETWORKS

One approach to dealing with temporal structures is the modified backpropagation network constructed by Elman (1990) (Fig. 6). The activity of the hidden layer is fed back as input to the network in the next iteration. This gives the network a form of temporal memory. The hidden layer can use nodes for coding the previous patterns presented to the network. Elman reached a good amount of success with this network.

The Elman network is capable of some evolvement of its categories, which makes it interesting. It can, for instance, encapsulate sequences by means of the recurrent connections. Further, these connections allow the network to form categories of categories. The main drawback of Elman networks, however, is that it – like the backpropagation network – is supervised. The self-organization that takes place in the hidden units is affected by this. The structures that are found are those that the supervisor has decided to be relevant. Any remaining hidden structures will not be found.

The procedure in which the network is used is to present a sequence consisting of several spatial patterns. The desired output is presented to the network and the weight adjustments are made. Typically, a lot of such sessions are required before the network stabilizes, which leads us to two objections: (1) The sessions are independent. The network is restarted from session to session and, hence, is not a continuous system in the sense previously used in this paper. (2) The learning is not incremental. All these limitations make the network unsuitable for our purposes.

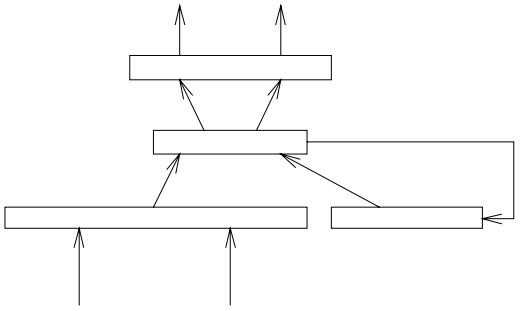

Figure 6: The architecture of an Elman network. The activity of the hidden layer is fed back to the input layer.

6 DISCUSSION

When we model biological organisms, we are striving to re-create a highly complex system. Neural networks are, however, generally simple. If we do not allow the network to evolve, it will remain simple and not reflect the complexity of the system it is intended to model. Most approaches to the extraction of more complex networks involve phylogenesis, but no ontogenesis. The popular approach to use genetic algorithms to evolve a neural architecture to solve a particular problem belongs to this field. With such approaches, we will construct highly advanced insectoids, but avoid the essence of cognition. The approach of this paper does not take us all the way to a cognitive system – far from it – but I hope that it is at least is in the right direction.

6.1 THE SYSTEM AS A SOCIAL ORGANIZATION

The cooperative learning model used for the suggested network invites us to view the network as a social organization. The synapses are the individuals, the neurons are groups of tightly associated individuals, and so on (Fig. 7). The synapses have two options available; either they cooperate with other synapses to activate or deactivate a neuron, or they defect. The group behaviour, embodied by the neuron, will highly affect the individual decision. Further, the group is associated with other groups, and their behaviour will affect

the group (neural) behaviour that again affects the individual. A social network will emerge.

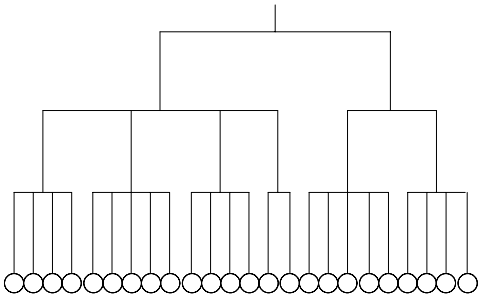

Figure 7: The network viewed as a social hierarchy. The circles at the bottom represent the synapses. The level above represents neurons. Even higher, we find groups of neurons, etc.

It has been shown that organizational structure in social groups can spontaneously emerge from the interactions of the group members (Glance and Huberman, 1993). This is especially evident in fluid organizations in which the individual can move within the organization. Obviously, the synapses in our neural network are unable to physically move. Higher up in the hierarchy, though, the case is different. A specific neuron can (functionally) move from one group to another, and an entire group of neurons can shift their cooperation or dissolve over time. The structure is still, however, somewhat constrained by the neural architecture.

To view the system as a social hierarchy illustrates one way in which several levels can emerge. By using stochastic learning automata to model the synaptic plasticity, a mean for indirect interaction between those levels has been introduced. These global interactions of numerous levels have been argued to be a primary characteristic of complex systems (Havel, 1993a; Havel, 1993b).

The synapses strive for optimizing their own behaviour, which is (for excitatory connections) to transmit signals only when the neuron subsequently becomes activated. This should be contrasted to many other learning algorithms for neural networks in which an external observer decides what the optimal parameter settings are. Such algorithms seldom have a perspective of more than one (emergent) level, which constrains the whole system (section 5).

6.2 Drawbacks and unsolved issues

As was pointed out previously, the actual implementation of temporal chunking is still obscure. One of the most important tasks of the future research of this model is to bring clarity into this issue. Is it an attribute of the synapses or the architecture? Is it a consequence of the spatial chunking? Should it be explicitly designed, or more implicitly to emerge from the self-organization?

The main problem when simulating this kind of model in computers is the computational cost. The approach requires a large number of neurons to be used. This means that an even larger number of synapses must be modelled. Further, the spontaneous activity as well as the stochastic learning automaton demands random number generation, which is also computational expensive. Without support from custom designed hardware, real-time simulation is not possible using the computers of today (or tomorrow). The interest in this approach lies therefore primarily in the theoretical domain.

Acknowledgements

I wish to thank my colleagues at Lund University Cognitive Science and the department of Computer Science for all their comments and especially Christian Balkenius. Further, I thank Matthew Hurst and Christer Johansson for inspiring discussions.

This work has been supported by the Swedish Council for Research in the Humanities and Social Sciences.

References

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169.

Balkenius, C. (1992). Neural mechanisms for self-organization of emergent schemata, dynamical schema processing, and semantic constraint satisfaction. *Lund University Cognitive Studies*, 14.

Balkenius, C. (1993). Natural intelligence for autonomous agents. In Svensson, B., editor, *International Workshop on Mechatronical Computer Systems for Perception and Action*, pages 181–188. Halmstad University, Sweden.

Barto, A. (1989). From chemotaxis to cooperativity: Abstract exercises in neuronal learning strategies. In Durbin, R. & Miall, C., editors, *The Computing Neuron*, pages 73–98. Addison-Wesley.

Carpenter, G. & Grossberg, S. (1993). Normal and amnesic learning, recognition and memory by a neural model of cortico-hippocampal interactions. *Trends in NeuroScience*, pages 131–137.

Edelman, G. M. (1987). *Neural Darwinism: The theory of Neuronal Group Selection*. Basic Books.

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

Glance, N. S. & Huberman, B. A. (1993). Organizational fluidity and sustainable cooperation. In Ghedira, K. & Sprumont, F., editors, *MAAMAW'93, Pre-proceedings of the 5th european*

workshop on "Modelling Autonomous Agents in a Multi-Agent World". Université de Neuchâtel.

Havel, I. M. (1993a). Artificial thought and emergent mind. In Bajcsy, R., editor, *IJCAI-93. Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 758–766. Morgan Kaufmann Publisher.

Havel, I. M. (1993b). Scale dimensions in nature. Technical Report CTS-93-03, Center for Theoretical Study, Prague.

Hinton, G. E. & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In Rumelhart, D. & McClelland, J., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol.1: Foundations*, pages 282–317. MIT Press.

Kohonen, T. (1989). *Self-organization and associative memory*. Springer-Verlag.

Narendra, K. S. & Thathachar, M. A. L. (1974). Learning automata—A survey. *IEEE Transactions on systems, man, and cybernetics*, SMC-4(4):323–334.

Pallbo, R. (1992). Neuronal selectivity without intermediate cells. *Lund University Cognitive Studies*, 13.

Pallbo, R. (1993). Visual motion detection based on a cooperative neural network architecture. In Sandewall, E. & Jansson, C. G., editors, *Scandinavian conference on artificial intelligence – '93*, pages 193–201. IOS Press.

Sato, H., Fox, K., & Daw, N. W. (1989). Effects on electrical stimulation of locus coeruleus on the activity of neurons in the cat visual cortex. *Journal of NeuroPhysiology*, 62(4):946–958.

Skarda, C. A. & Freeman, W. J. (1987). How brains make chaos in order to make sense of the world. *Behavioral and Brain Sciences*, 10:161–195.

Squatrito, S., Trotter, Y., & Poggio, G. F. (1990). Influences of uniform and textured backgrounds on the impulse activity of neurons in area V1 of the alert macaque. *Brain Research*, 536:261–270.

Tsetlin, M. L. (1973). *Automaton Theory and Modeling of Biological Systems*, volume 102 of *Mathematics in science and engineering*. Academic Press.

Werner, G. (1987). Cognition as self-organizing process (Commentary to How brains make chaos in order to make sense of the world, Skarda & Freeman). *Behavioral and Brain Sciences*, 10:183.

White, H. (1990). Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural Networks*, 3:535–549.