

# ARTIFICIAL GRAMMAR PROCESSING IN SPIKING NEURAL NETWORKS

Philip Cavaco, Baran Çürüklü

*School of Innovation Design and Engineering, Mälardalen University, Västerås, Sweden*  
pco07001@student.mdh.se, baran.curuklu@mdh.se

Karl Magnus Petersson

*Max Planck Institute for Psycholinguistics & Donders Institute for Brain, Cognition and Behaviour,  
Radboud University of Nijmegen, Nijmegen, Netherlands*  
Karl-Magnus.Petersson@mpi.nl

**Keywords:** Artificial grammar recognition, Structured sequence processing, Spiking neuron model, Minicolumn architecture, Reber grammar.

**Abstract:** In this paper we explore the feasibility of artificial (formal) grammar recognition (AGR) using spiking neural networks. A biologically inspired minicolumn architecture is designed as the basic computational unit. A network topography is defined based on the minicolumn architecture, here referred to as nodes, connected with excitatory and inhibitory connections. Nodes in the network represent unique internal states of the grammar's finite state machine (FSM). Future work to improve the performance of the networks is discussed. The modeling framework developed can be used by neurophysiological research to implement network layouts and compare simulated performance characteristics to actual subject performance.

## 1 INTRODUCTION

Cognitive neuroscience approaches the brain as a cognitive system, functionally conceptualized in terms of information processing. More specifically, a physical system can be viewed as an information processing device when (a subclass of) its physical states are representational and state transitions can be viewed as a process operating on these states by implementing operations on the corresponding representational structures. A common framework for describing cognitive systems in cognitive neuroscience is Marr's three descriptive levels of analysis: the computational, the algorithmic, and the implementational (Marr, 1982). Cognitive neuroscience typically works with a modified perspective including simultaneously, perspectives on cognition, learning, and development. However, the underlying ideas are essentially the same. Within this framework, the three descriptive levels are generalized to: 1) the cognitive or computational level - a formal theory of structured cognitive states and its transition system; 2) the dynamical system level - given a formal cognitive theory, a state-space is specified and operations are formulated in terms of a dynamical system that embeds the cognitive level specification; 3) the implementation level - this level spec-

ifies the physical hardware implementation of the dynamical system, for example a neural network specified in terms of network topology, synaptic and neuronal dynamics. This framework is naturally extended to developmental learning system (Petersson, 2008). A developmental learning system can be conceptualized as a system of coupled dynamics: a dynamics for processing of information in interaction with one (or several) dynamics for learning, memory, and development. We thus arrive at a dynamic conceptualization of a cognitive learning system  $D = \langle \text{functional architecture, representational dynamics, learning dynamics} \rangle$  in terms of: 1) *functional architecture*: a specification of the structural organization of the system (e.g., network architecture); 2) *representational dynamics*: a specification of a state-space  $\Omega$  carrying/representing information, and dynamical principles,  $T : Mx\Omega \times \Sigma \rightarrow \Omega \times \Lambda$ , governing the active processing of information (e.g., evolution of membrane potentials and action potential generation); and 3) *learning dynamics*: a configuration space  $M$ , which includes a specification of learning (adaptive) variables/parameters for memory formation and information storage in a general sense (e.g., synaptic parameters), as well as dynamical principles, a "learning algorithm",  $L : Mx\Omega \rightarrow M$ , that govern the tempo-

ral evolution of the learning variables in configuration space (e.g., Hebbian learning). The temporal evolution of the adaptive parameters depends on the active processing of information and the learning dynamics is typically, but not necessarily, conceived of as taking place on a slower (or longer) characteristic time-scale than that of the representational dynamics. In this paper we will focus on the feasibility of implementing a representational dynamics that corresponds to processing of structured sequences in a minicolumn spiking network architecture.

A generic and fundamental problem in cognition is how the processing of structured sequences is organized. A paradigmatic example is human natural language. In an approach addressing the problem of structured sequence processing, we illustrate how to integrate levels of analysis within a framework of adaptive dynamical systems. We use grammar or syntax processing as a concrete example throughout, although much is still unknown about language processing in the human brain (Jackendoff, 2002). In this paper, we combine computer science and cognitive neuroscience methods to explore the paradigm of artificial grammar recognition (AGR). We show that a biologically inspired spiking neuron architecture can be built to perform AGR. It can classify strings belonging to a specific regular grammar, here called the Reber grammar (Reber, 1967). The work presented here extends work by Petersson et al. (2004) on the simple recurrent network (SRN) architecture which showed that the capacities of SRNs are sufficient to recognize strings generated from the Reber grammar at high performance levels (Petersson et al., 2005). In this context, we note that the dynamical systems framework, in particular spiking neural networks, naturally lends itself to a description of event-driven asynchronous information processing devices, which is of central importance because the brain appears to be an asynchronous processing system (Petersson, 2008).

Our work provide new tools for current and future cross-disciplinary and translational research combining behavioral (Folia et al., 2008; Forkstam et al., 2008), neurophysiological (Petersson et al., 2004; Forkstam et al., 2006; Uddén et al., 2008) and simulated network experiments (Petersson et al., 2005; Cavaco et al., 2009) as well as theoretical investigations (Petersson, 2005; Petersson, 2008). The AGR task given to the simulated network in this study is analogous to experimental tasks given to humans in both behavioral (Uddén et al., 2009) and functional magnetic resonance imaging (fMRI) experiments (Petersson et al., 2004; Forkstam et al., 2006).

Background research motivating modeling decisions made in this work are presented in Section 2.

Section 3 presents the details of the model layers: the input layer, which transduces input strings to spiking activity; and the basic processing unit, a biologically inspired minicolumn architecture and the network topology. Results are presented and network performance is evaluated in Section 4. Finally, future research is discussed in Section 5.

## 2 BACKGROUND

Simple recurrent networks (SRNs) with analogue neurons can identify strings as belonging to a Reber grammar or not with a high accuracy (Petersson et al., 2005). Petersson et al. (2004) analyzed the properties of the Reber grammar mathematically. Then SRNs were created to recognize strings generated by the finite state machine (FSM; i.e., recognizer/generator) corresponding to the Reber grammar, (Fig. 1). Strings generated with this FSM begin with the symbol #, followed by a number of other characters from the set /M, V, X, T, R/, the end symbol #.

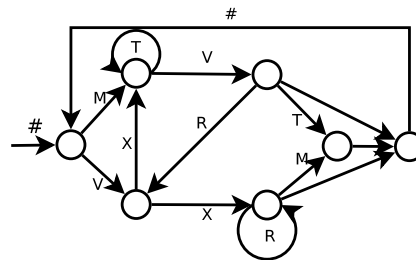


Figure 1: The Reber machine used in this paper: an information processing device formulated within the classical Church-Turing framework.

In general, generic sequence recognition requires some form of on-line (short-term) memory that supports sequence look-back capacity because both adjacent and non-adjacent (long-distance) dependencies have to be processed in a meaningful way for successful and efficient sequence recognition. Petersson et al. (2004) found that a 3-symbol sliding look-back window is "logically" sufficient to achieve optimal performance in predicting the next symbol from a string generated by the Reber grammar (Petersson, 2005), (Fig. 1). Moreover, retention of two concepts has been demonstrated in a spiking neuron network model of short-term memory that focused on properties of recurrent connectivity to sustain activity within the network (Macoveanu et al., 2006). The connectivity strength in recurrent networks have an effect on short-term (working) memory properties and Tegner et al. suggested that neurons with recurrent connections excite each other to retain a memory in the form

of a dynamic attractor (Tegnér et al., 2002). A concept similar to this is used in in this paper.

### 3 MODEL

The minicolumn architecture developed in this paper uses the combined activity of all of the excitatory neurons within a given minicolumn to represent a recent input symbol. This model is designed to retain activity resulting from presentation of a symbol for the duration of the two subsequent symbols from the grammar, using active storage (Zipser et al., 1993). This produces a two symbol look-back capability of the minicolumn architecture. A third look-back symbol is achieved through activation of sublayers within a network of minicolumn architectures. The activation of these sublayers indicate that a specific substring has been observed by the relevant parts of the network. A similar concept of fading memory has been implemented in other network models (Maass et al., 2007; Sandberg et al., 2001). In this context, memory decay implies that the influence of any specific segment of the input stream on later parts of the output stream becomes negligible when the length of the intervening time interval is sufficiently large (Maass et al., 2007).

Using biologically inspired spiking neurons, we designed a network that recognizes strings belonging to the Reber grammar. The architecture is divided into an input layer and a recognition layer. The input layer converts (transduces) input symbols into spiking activity. The recognition layer is a network of minicolumn architectures designed to retain the rules of the grammar and match the input activity to the rules. Our spiking minicolumn architecture was implemented in the Neural Simulation Tool (NEST) (Gewaltig and Diesmann, 2007), which provides neuron models as well as a framework for simulating large neural networks, and was used for all modeling and simulation described in this paper.

The recognition layer is designed to accept sequences generated by the Reber grammar defined in (Fig. 1). The network attempts to minimize the number of nodes (minicolumn architectures), in the network, needed to track a string’s position in the FSM corresponding to the grammar, shown in Section 3.3. This network connects minicolumns in a tree structure designed to recognize subparts of sequences. The function of this part of the network is that activity in nodes located in lower positions in the tree signals the presence of substrings in the input sequences of increasing length. The first level of the tree is connected to the input layer (Fig. 2).

### 3.1 Input Layer

The strategy for translating character strings into spiking activity was inspired by sensory transduction mechanisms in neural systems. The 5 symbols in Reber strings plus the end of string symbol  $\{\#,M,T,R,X,V\}$  are treated as separate inputs. Each symbol causes excitatory activity from a different input node (Fig. 2). The input string is presented to the network one symbol at a time, only the input node of the current symbol is active, which thus corresponds to a lexical detector/retriever in the human brain (Jackendoff, 2002). These input nodes are in turn connected to minicolumns which are sensitive to, and “listen” for, specific input patterns and begin spiking once an input is present. In the simulation, input is considered a representation of a particular sound pattern as in, for example a word. Input could also be considered as visual or somatosensory input without loss of generality.

The input layer is implemented as 6 direct current (DC) generators in NEST. Each is mapped to a specific input symbol  $\{\#,M,X,T,V,R\}$ . When a symbol is presented to the network the corresponding DC generator is set to a positive amplitude. Only one generator is active at a given point in time and all other generators have their amplitude set to zero. DC generators are in turn connected to input nodes in the recognition layer. In the case where the input layer is presented with the first character from the string #MTVT# (Fig. 2). The DC generator representing # is activated and all other generators are silent. The resulting current excites the connected minicolumn sensitive to the symbol #. Activity will then propagate to all nodes connected to this minicolumn.

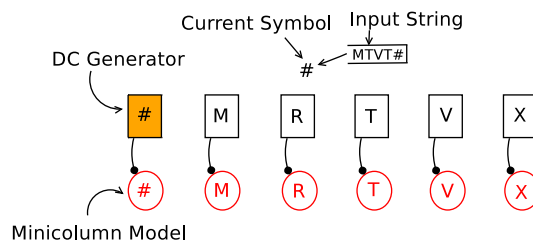


Figure 2: The input layer receiving the string “#MTVT#”. When the symbol # is presented the corresponding DC generator, the highlighted square indicated by #, is activated by setting its output rate. This causes the connected minicolumn to become active and inject activity the rest of the network

### 3.2 Minicolumn Model

Biologically inspired minicolumn models are the main processing units of the network. Each minicol-

umn is designed with 100 integrate and fire neurons, 80% excitatory and 20% inhibitory. These proportions are based on neurophysiological data (Buxhoeveden and Casanova, 2002; Çürüklü and Lansner, 2003; Çürüklü and Lansner, 2002; Çürüklü and Lansner, 2001; Çürüklü and Lansner, 2005). The models studied in this paper use local connections within the minicolumn to create recurrent activity to represent the presence of input. The model neurons were implemented as neurophysiologically inspired leaky integrate and fire models. Generally we used plausible parameters in the network model so that the performance of the model can be related to real neurophysiological investigations, for example electroencephalogram (EEG) or functional magnetic resonance imaging (fMRI) experiments. Moreover, the model used in this paper uses networks to represent the hierarchical connectivity distribution observed in the primate brain (Felleman and Van Essen, 1991), in which concepts are learned and stored and combinations of these concepts can be reused, for example, to identify more abstract entities.

The first goal of this research was to achieve degrading persistent spiking within the minicolumns. In other words, following presentation of input, the minicolumn should continue spiking from internal recurrent activity of excitatory neurons for a period of two symbol presentations, specified to be 1000 ms in the model (each symbol is presented for 500ms). At that time, local inhibition overcomes the excitatory activity and the neurons return to their unexcited activity rates. This effectively provides a 2 symbol look-back sliding window memory mechanism. This time scale was chosen to be biologically plausible based on behavioral and neurophysiological experiments with humans. For example, symbol sequences from the Reber grammar have been presented for 300ms each with a 300ms inter-symbol-interval (Forkstam et al., 2006), and auditory stimuli has been presented to human subject for 500ms with 5ms rise/fall time and (Alain et al., 2001).

Leaky integrate and fire neurons approximate the behavior of biological neurons using the mathematical model of Rotter and Diesmann (Rotter and Diesmann, 1999). This is a standard approximation which is computationally efficient and allows simulation of large networks (Lobb et al., 2005). The leaky integrate and fire neurons have a rest/reset membrane potential of  $-75\text{mV}$  and a firing threshold at  $-55\text{mV}$ . An absolute refractory period ( $T_{\text{Ref}}$ ) of 5ms is used. This is necessary to achieve activity rates at a biologically plausible level ( $\sim 35\text{Hz}$ ).

Each excitatory neuron is connected to  $\sim 40\%$  of other excitatory neurons, and  $\sim 25\%$  of inhibitory

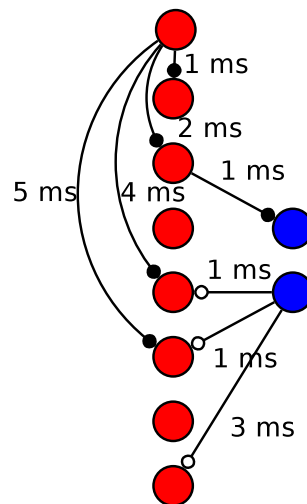


Figure 3: A simplified model of connections within the minicolumn. Numbers indicate base delay, in ms, along the connection representing spacial locality of the neurons. These values are later offset by a random amount to simulate lateral position of neurons and indirect axon paths. Filled circles represent excitatory connections, empty circles inhibitory. Each connection type is shown from only one neuron.

neurons. Each inhibitory neuron is connected to  $\sim 15\%$  of excitatory neurons. A simplified view of local minicolumn connections (Fig. 3). Each type of connection (excitatory-excitatory, excitatory-inhibitory, inhibitory-excitatory) from one neuron is shown. When connections are made, each neuron is connected to a set of neurons randomly selected from the available population within the minicolumn.

Given an excitatory neuron  $x$  from the collection of all excitatory neurons  $Excite$ , the set of excitatory neurons  $x$  will connect to is obtained by randomly selecting 40% of remaining excitatory neurons. For excitatory-inhibitory connections, 25% of the inhibitory neurons are selected. The randomized connectivity procedure ensures that the connections in each minicolumn are not identical.

Spatial locality is built into the model. Neurons are laid out as in (Fig. 3), with the inhibitory neurons placed in the middle of the excitatory neurons. For two neurons,  $x$  and  $y$ , the distance between them,  $dist$ , is calculated with the neurons position in an array. A circular neuron layout where angles between neurons introduce variation in the connection delay, has previously been used (Compte et al., 2000). For two excitatory neurons,  $x$  and  $y$ , in the array  $Excite$ , the distance between them,  $dist$ , is calculated as in (Eq. 1). For an excitatory neuron,  $x$ , and a inhibitory,  $i$ , in the array  $Inhib$  the distance between the neurons,  $dist$ , is calculated as in (Eq. 2).  $INHIB\_ORIGIN$  is the position on the excitatory array where the first inhibitory neuron

lies. For example, in (Fig. 3)  $INHIB\_ORIGIN = 3$ .

$$dist = abs(Excite.index(y) - Excite.index(x)) \quad (1)$$

$$dist = abs(Excite.index(x) - (INHIB\_ORIGIN + Inhib.index(i))) \quad (2)$$

A random delay,  $offset$ , in the range of  $\pm 2.5ms$  is applied to each connection to simulate lateral location of the neuron and non-direct axon paths,  $offset$  is calculated by (Eq. 3). Total connection delay,  $del$ , is calculated by (Eq. 4).

$$offset = 2 \times 2.5 \times rand() - 2.5 \quad (3)$$

$$del = dist + offset \quad (4)$$

$$del_f = del \times DELAY\_FACTOR \quad (5)$$

The final calculated delay is scaled by a  $DELAY\_FACTOR$ . This is used to scale the size of each unit of delay. In the brain action potentials propagate a distance of  $\sim 10cm$  within 10-12ms (Aboitiz et al., 1992). The final delay,  $del_f$  for a connection is calculated as indicated in 5.

Background noise is added by simulated Poisson distributed spikes at 10Hz activity from 200 neurons. This causes the membrane potential to remain around -65mV when no other input is present. Each neuron receives a separate Poisson spike train.

Connection weights are scaled experimentally to achieve activity characteristics comparable to typical neurobiological findings. Excitatory weights are scaled so that between 1 to 4 spikes arriving within a 4ms time window are required to cause a spike. Connection weights are set as follows: excitatory 30, inhibitory -44, input 18, noise 21. These settings produce excited activity of  $\sim 50Hz$  and recurrent activity for  $\sim 1000ms$  after external stimulus is removed. Each minicolumn is configured to be excited by a single input symbol or two other minicolumns with approximately 50Hz spiking activity.

Recurrent activity of approximately 50Hz within each minicolumn indicates that the symbol or subsequence the minicolumn is sensitive to was present in the input sequence. This level of retention provides a short symbol look-back. Temporal information of the character order is gradually lost when two minicolumns are active at the same time. To retain information about sequence order a hierarchical network of minicolumns is utilized. Increasingly deep layers are activated by increasingly long substrings generated by the FSM.

Inter-columnar connectivity is defined in the network architecture in the following manner. Inter-excitatory connections connect each excitatory neuron in the source minicolumn to 10% of the excitatory neurons in the target minicolumn. Inter-inhibitory connections connect each excitatory neuron to 20% of inhibitory neurons in the target. This is consistent with connection densities in biological models (Çürüklü and Lansner, 2003; Çürüklü and Lansner, 2002; Çürüklü and Lansner, 2001; Çürüklü and Lansner, 2005). A larger minimum delay of 3ms is applied to all long-distance interconnections between columns to represent the increased distance between minicolumns.

### 3.3 Network

The objective of the network was to use a small number of nodes to recognize substrings by limiting lower level nodes, shown in (Fig. 4), which uses only 20 network nodes.

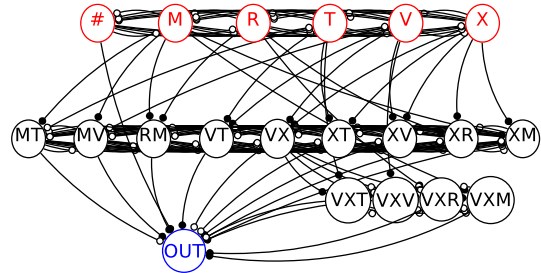


Figure 4: A network with a minimized set of minicolumns still able to recognize strings in the grammar.

Nodes are selected by traversing the FSM and adding nodes which uniquely identify states. For example MV, VT, VX indicate the edge into and out of states of the FSM. It was found that most trigrams were not necessary. The trigrams which do have nodes in the network cover those parts of the FSM where two edges can be taken to produce the same symbol, namely X, that is the non-deterministic parts of the FSM.

The output node OUT receives input from all nodes which represent substrings occurring at the end of a string, namely the nodes representing #, RM, VT, XM, VXR, VXM, MV. This introduces a great deal of activation of the output node but is compensated with inhibition.

Inter-inhibition connectivity is made between all nodes of the same level. This forces only one minicolumn, at each level, to remain fully active. Only the minicolumn currently receiving input from higher levels should be fully active. However, this does not remove the extra excitation from OUT. Inter-inhibition connectivity is made from nodes which

cannot be the end of a string to OUT, including MT, VX, VXT, XT. For example, MT can never occur at the end of a grammatical string so the node representing this bigram has inter-inhibition connectivity to OUT. This network results in a continuous low level of activity in OUT during string presentation.

## 4 RESULTS

Simulations were performed to determine the feasibility of the network design for artificial grammar recognition (AGR). Grammatical, non-grammatical, and random sequences of 100 symbols, are presented to the network. The grammatical and non-grammatical and random sequences all contain 17 strings. Non-grammatical strings were manually created by altering individual symbols in the corresponding grammatical string. This is consistent with how strings are constructed in behavioral and neurophysiological experiments (Folia et al., 2008; Forkstam et al., 2006; Petersson et al., 2004; Uddén et al., 2008). Random strings such as, MVVRVV, are generated by selecting one symbol from the alphabet {#, M, V, T, R, X}, where each symbol has equal probability of being selected for each position. Strings generated this way are generally non-grammatical, some empty strings are also produced, for example ## which does not contain any symbols. These are removed in post processing since double # will cause the output node to activate. Strings must have at least 2 symbols between # symbols, since the smallest string generated by the grammar contain 2 symbols.

Random non-grammatical strings are easily identified as non-grammatical by the network when no valid substrings exist. A string such as, #MTRT#, is more difficult to identify as non-grammatical since only one symbol is incorrect. The network can falsely recognize these strings if the last substrings are grammatical. The network being able to recognize these types of strings as non-grammatical provide good support for its use in future computational investigations.

Network activity was analyzed through spike train analysis. The spike activity of all neurons in a given minicolumn is collected throughout the simulation. Histograms are then plotted to analyze spiking rates, (Fig. 5). The input sequence used is set as the x-axis of the histogram plot so that activity can be correlated to characters being presented. Recognition is defined as the output node having an average activity of 48Hz for the 500 ms following the end of string symbol # (Fig. 5(a)).

The results of the simulations are shown in Ta-

ble 1. The network endorses 16 of 17 grammatical strings, 10 of 17 non-grammatical strings, and 7 of 17 random non-grammatical strings. These results show that the network is able to discriminate between grammatical and non-grammatical strings. A high number of false positives exist in the non-grammatical and random non-grammatical sequences. The strings recognized in the random sequence contain substrings similar to valid substrings. The sequence MX appears in a number of places in the random sequence, this is the inverse of the correct sequence XM. Presentation of these inverse substrings, such as MX, cause activation within the network in the nodes representing the grammatical substring, such as XM.

Three activity graphs show the activity in the output minicolumn from the network from the first 30 symbols of the grammatical, non-grammatical, and random input sequences (Fig. 5). The output node is active when the input string contains grammatical substrings which can end a string.

Table 1: The results of presenting the input sequences to the network.

	Network	
	Strings Recognized	Percentage of strings recognized as grammatical
Grammatical	16 of 17	94.12%
Non-Grammatical	10 of 17	58.82%
Random	7 of 17	41.18%

## 5 DISCUSSION

Cognitive neuroscience approaches the brain as an information processing device and we have outlined some relevant aspects of this concept as applied to the problem of structured sequence processing in a minicolumn network architecture. Generally, cognitive brain functions can be understood as the outcome of a developmental learning system  $D = \langle \text{functional architecture, representational dynamics, learning dynamics} \rangle$ . In this paper we focused on creating a functional architecture with a representational dynamics and we have implicitly indicated how it is possible to integrate Marr's levels of analysis within the framework of dynamical systems. General dynamical system theory is obviously too rich as a framework for formulating explicit models of cognitive brain functions. For example, it turns out that for any given state-space one can find a universal dynamical system whose traces will generate any dynamics on the

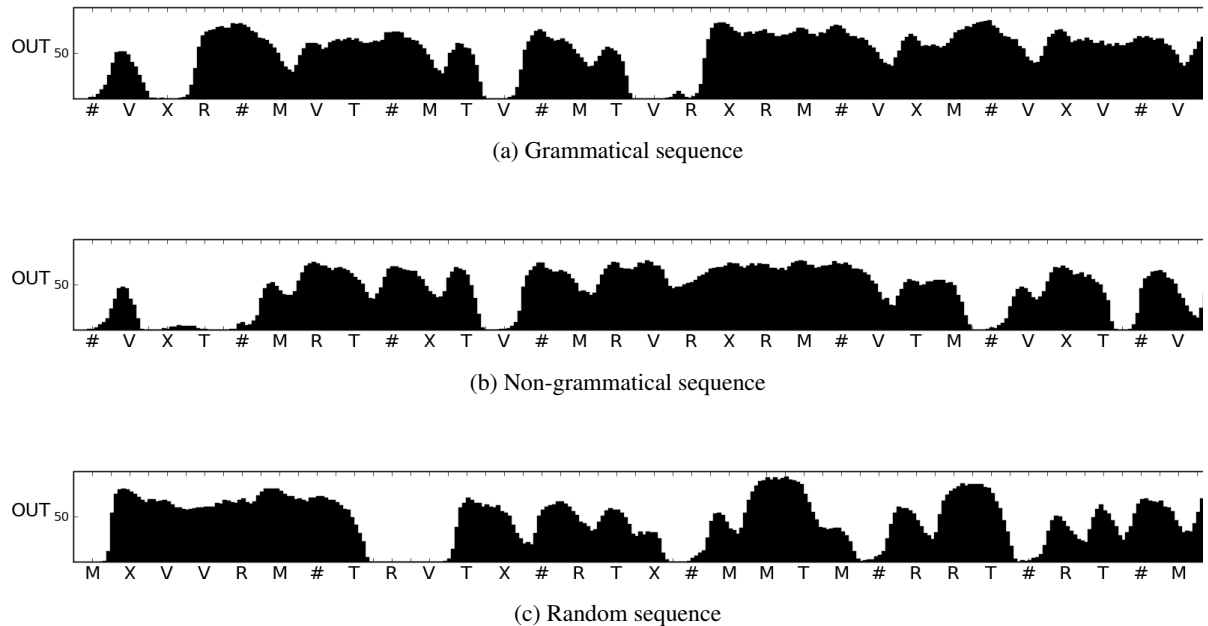


Figure 5: Activity from the output node for the first 30 symbols of the three input sequences a) grammatical, b) non-grammatical, c) random.

state-space (Lasota and Mackey, 1994). Thus, what is needed is a specification of cognitively relevant constraints as well as processing principles relevant for the neurobiological networks subserving information processing in the brain. Our spiking network architecture is an example of the dynamical systems approach in cognitive neuroscience that offers a descriptive framework for analyzing asynchronous event-driven devices, which is essential in understanding real neural systems. Finally, it is clear that our work so far on artificial grammar recognition is in its infancy with respect to a more profound understanding of sequentially organized, structured processing in the brain.

We have demonstrated that artificial grammar recognition is feasible in spiking neural networks. It is also evident that this paper only touches on this complex topic. The original inspiration for the work presented here was to create models based on specific neurophysiological knowledge and neocortical theories. The ability to meet these goals was limited mainly by two factors. Firstly, the technical hurdles faced are vast. Secondly, the lack of a unified theory of neural structure based on neuroimaging and neurophysiological experiments is currently not available. Future directions of the current work are to explore flexibility/adaptability and learning in spiking minicolumn network architectures.

In this paper, the network is created with an inherent preference for the Reber grammar. A natural

next step would be to begin with a more generic form of the network with a different number of nodes (i.e., minicolumns) than in the current network. For example, the nodes could be completely connected to one another and made dynamic/adaptive. The characteristics of various self-organizing learning schemes, implemented as synaptic adaptation and pruning (e.g., removing a connection when the synaptic weight becomes negligibly small), can then profitably be investigated in order to study if and how functional networks emerge as a result of exposure to an acquisition set of training items and then test the network on a disjoint classification set and compare with human performance as well as relevant neurophysiological measures. Another strategy would be to use genetic programming to evaluate an optimal network layout and parameter settings for the network.

Fundamental issues to address in future work also include determining what might be a relevant state-space, with associated dynamics, in order to gain a theoretical (or mathematical) understanding of active information processing in general spiking network architectures. Moreover, issues related to what might serve as relevant prior constraints as well as the characteristics of the adaptive machinery that allows all normal human individuals to acquire a native language robustly and in a relatively short period of time need to be addressed. These are essentially questions about the nature of the neural code, the dynamical

variables that serve as information carriers, and its adaptive characteristics.

## ACKNOWLEDGEMENTS

We thank Julia Uddén for helpful comments on earlier drafts of this manuscript and Christian Forkstam for support throughout the project.

## REFERENCES

- Aboitiz, F., Scheibel, A. B., Fisher, R. S., and Zaidel, E. (1992). Fiber composition of the human corpus callosum. *Brain Research*, 598:143–153.
- Alain, C., Arnott, S. R., and Hevenor, S. (2001). "What" and "where" in the human auditory system. *PNAS*, 98:12301–12306.
- Buxhoeveden, D. P. and Casanova, M. F. (2002). The minicolumn hypothesis in neuroscience. *Brain*, 125:935–951.
- Cavaco, P., Çürüklü, B., and Petersson, K. (2009). Artificial grammar recognition in neural networks. Master's thesis, Mälardalen University.
- Compte, A., Brunel, N., Goldman-Rakic, P. S., and Wang, X. J. (2000). Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex*, 10(9):910–923.
- Çürüklü, B. and Lansner, A. (2001). Spike and burst synchronization in a detailed cortical network model with i-f neurons. In G. Dorffner, H. Bischof, K. H., editor, *Proc. of the International Conference on Artificial Neural Networks*, pages 1095–1102, Vienna, Austria. Springer-Verlag.
- Çürüklü, B. and Lansner, A. (2002). An abstract model of a cortical hypercolumn. In L. W., JC, R., K, F., S-Y, L., and X, Y., editors, *Proc. of the 9th International Conference on Neural Information Processing*, pages 80–85, Singapore. IEEE.
- Çürüklü, B. and Lansner, A. (2003). Quantitative assessment of the local and long-range horizontal connections within the striate cortex. In *Special Session on "Biologically Inspired Computational Vision" at the 2nd Int. Conf. on Computational Intelligence, Robotics and Autonomous Systems*, Singapore. IEEE.
- Çürüklü, B. and Lansner, A. (2005). A model of the summation pools within the layer 4 (area 17). *Neurocomputing*, (65-66):167–172.
- Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex*, 1:1–47.
- Folia, V., Uddén, J., Forkstam, C., Ingvar, M., Hagoort, P., and Petersson, K. M. (2008). Implicit learning and dyslexia. *Ann. N. Y. Acad. Sci.*, 1145:132–150.
- Forkstam, C., Elwér, A., Ingvar, M., and Petersson, K. M. (2008). Instruction effects in implicit artificial grammar learning: a preference for grammaticality. *Brain Research*, 1221:80–92.
- Forkstam, C., Hagoort, P., Fernandez, G., Ingvar, M., and Petersson, K. M. (2006). Neural correlates of artificial syntactic structure classification. *Neuroimage*, 32:956–967.
- Gewaltig, O. and Diesmann, M. (2007). Nest (neural simulation tool). *Scholarpedia*, 2(4):1430.
- Jackendoff, R. (2002). *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press.



- Lasota, A. and Mackey, C. (1994). *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*. Springer.
- Lobb, C. J., Chao, Z., Fujimoto, R. M., and Potter, S. M. (2005). Parallel event-driven neural network simulations using the Hodgkin-Huxley neuron model. In *PADS '05: Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation*, pages 16–25, Washington, DC, USA. IEEE Computer Society.
- Maass, W., Joshi, P., and Sontag, E. D. (2007). Computational aspects of feedback in neural circuits. *PLoS Comput Biol*, 3(1):e165.
- Macoveanu, J., Klingberg, T., and Tegner, J. (2006). A biophysical model of multiple-item working memory: a computational and neuroimaging study. *Neuroscience*, 141:1611 – 1618.
- Marr, D. (1982). *Vision: a computational investigation into the human representation and processing of visual information*. W.H. Freeman, San Francisco ; New York.
- Petersson, K., Forkstam, C., and Ingvar, M. (2004). Artificial syntactic violations activates Broca's region. *Cognitive Science*, 28:383–407.
- Petersson, K. M. (2005). On the relevance of the neurobiological analogue of the finite-state architecture. *Neurocomputing*, 65-66:825–832.
- Petersson, K. M. (2008). On cognition, structured sequence processing, and adaptive dynamical systems. volume 1060, pages 195–199. AIP.
- Petersson, K. M., Grenholm, P., and Forkstam, C. (2005). Artificial grammar learning and neural networks. *Proceeding of the Cognitive Science Society 2005*.
- Reber, A. S. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behavior*, 6:855–863.
- Rotter, S. and Diesmann, M. (1999). Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. *Biological Cybernetics*, 81:381–402.
- Sandberg, A., Lansner, A., and Petersson, K. M. (2001). Selective enhancement of recall through plasticity modulation in an autoassociative memory. *Neurocomputing*, 38-40:867–873.
- Tegner, J., Compte, A., and Wang, X. J. (2002). The dynamical stability of reverberatory neural circuits. *Biological Cybernetics*, 87(5-6):471–481.
- Uddén, J., Araujo, S., Forkstam, C., Ingvar, M., Hagoort, P., and Petersson, K. (2009). A matter of time: Implicit acquisition of recursive sequence structures. In *Proceedings of the Cognitive Science Society*.
- Uddén, J., Folia, V., Forkstam, C., Ingvar, M., Fernandez, G., Overeem, S., van Elswijk, G., Hagoort, P., and Petersson, K. M. (2008). The inferior frontal cortex in artificial syntax processing: an rTMS study. *Brain Research*, 1224:69–78.
- Zipser, D., Kehoe, B., Littlewort, G., and Fuster, J. (1993). A spiking network model of short-term active memory. *Journal of Neuroscience*, 13:3406–3420.