



Symbol manipulation and rule learning in spiking neuronal networks

Chrisantha Fernando ^{a,b,c,*}

^a Department of Informatics, University of Sussex, Falmer, Brighton, BN1 9RH, London, UK

^b MRC National Institute for Medical Research, The Ridgeway, Mill Hill, London, UK

^c Collegium Budapest, Institute for Advanced Study, Szentháromság u. 2, H-1014 Budapest, Hungary

ARTICLE INFO

Article history:

Received 19 August 2010

Received in revised form

9 January 2011

Accepted 10 January 2011

Available online 13 January 2011

Keywords:

Spiking neural network

Neuronal replicator hypothesis

Learning classifier systems

Spike-time-dependent plasticity

ABSTRACT

It has been claimed that the productivity, systematicity and compositionality of human language and thought necessitate the existence of a physical symbol system (PSS) in the brain. Recent discoveries about temporal coding suggest a novel type of neuronal implementation of a physical symbol system. Furthermore, learning classifier systems provide a plausible algorithmic basis by which symbol re-write rules could be trained to undertake behaviors exhibiting systematicity and compositionality, using a kind of natural selection of re-write rules in the brain. We show how the core operation of a learning classifier system, namely, the replication with variation of symbol re-write rules, can be implemented using spike-time dependent plasticity based supervised learning. As a whole, the aim of this paper is to integrate an algorithmic and an implementation level description of a neuronal symbol system capable of sustaining systematic and compositional behaviors. Previously proposed neuronal implementations of symbolic representations are compared with this new proposal.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

In a highly influential and controversial paper, Fodor and Pylyshyn provided an argument for the existence of a physical symbol system (PSS) in the brain (Fodor and Pylyshyn, 1988). A simple way to understand what is meant by a PSS is to map the cognitive concepts onto concepts from chemistry. This analogy should not be taken too far, and is best used as an intuition pump to broaden the way we think about symbol processing. We do not wish to claim that a neuronal physical symbol system is isomorphic to a chemical one, and so at each stage we discuss the differences as well as the similarities.

Chemistry deals with molecules that are composed of atoms. Structural relations between atoms define a molecule. There is a combinatorial syntax, i.e. a set of chemical structural constraints such as valance, charge, etc., that determine how atoms can legally join together to make molecules.

Furthermore, the structure of a molecule has information about its *chemical function or reactivity*, and this is systematically related to the function of its parts, e.g. the structure of the benzene ring *means* that it will react in a certain way in a given environment, and the fact that it has a methyl group *means* that this reactivity will be changed in a systematic way in that

environment. This is a kind of internal semantics. Internal semantics deals with how a symbol contains information about the reactivity within the symbol system itself. It is also called compositionality, and we will discuss it shortly, but before this, we should highlight that there is another kind of semantics, which we call external semantics. External semantics deals with how a symbol contains information about the outside world (i.e. its semantic interpretability) and it is easiest to consider this for biochemical systems; here molecules can clearly be seen to also have semantic content, i.e. parts of the molecule may confer information about the environment external to the symbol system. For example, the conformation of hemoglobin can confer information about the oxygen saturation, or the concentration of a cell signaling molecule can confer information about glucose concentration.

Fodor and Pylyshyn proposed that there is a PSS implemented in neuronal structures that has similar properties to a molecular symbol system. To understand why, it is useful to compare chemical experiments with human language and thought. The properties of atoms and molecules described above give chemistry a special set of macroscopic characteristics. For example, chemistry is productive. The capacity for chemical reactivity is unlimited. Indefinitely many molecules can be produced allowing indefinitely many reactions. This is possible with only a finite set of distinct atomic types. Therefore, an unbounded set of chemical structures must be non-atomic. In the same way, an indefinite number of propositions can be entertained, or sentences spoken. This is known as the productivity of thought and language,

* Correspondence address: Department of Informatics, University of Sussex, Falmer, Brighton, BN1 9RH, London, UK.

E-mail address: ctf20@sussex.ac.uk

therefore neuronal symbols must have the same capacity for being combined in unlimited ways.

Secondly, chemistry is systematic, that is the capacity for atoms to be combined in certain ways to produce some molecules is intrinsically connected to their ability to produce others. Consider how a chemist might learn chemistry. There are rules of thumb that help a chemist to guess how atoms will form a molecule, and how that molecule will react based on its structure. A chemist does not learn just a list of valid molecules or reactions. In the same way, there is systematicity in language, e.g. the ability to produce or understand a sentence is intrinsically connected with the ability to produce and understand other sentences. This is because there is systematicity in the way that the physical symbols responsible for language can form symbol structures and there is systematicity in the way that this structure determines the reactivity of these structures. Because of this structure, languages need not be learned by learning a phrasebook. Languages have syntax. No English speaker can say A loves B, but not be able to say B loves A.

Thirdly, we have already discussed how the same atom makes approximately the same contribution to each molecule in which it occurs. This means that there is systematicity in reactivity (semantics) as well as in structure (syntax). This is known as compositionality. In the same way, lexical items in sentences have approximately the same contribution to each expression in which they occur. This is a property of internal meanings (semantics), i.e. what a structure means in terms of function.

The phenomena of language and thought imply a neuronal physical symbol system, i.e. a system with all the properties described above, in the same way as the phenomena of chemistry imply the existence of atoms and molecules. However, there are *extra* properties required of the PSS in cognition compared to the PSS of chemistry. The most important is the fact that cognition includes the capacity to *learn* an appropriate PSS, not just to implement it. The fact that children can learn and manipulate explicit rules (Clark, 1991; Karmiloff-Smith, 1996) implies the existence of a neuronal physical symbol system capable of forming structured representations (analogous to molecules) and learning rules for operating on these representations (analogous to reactions of molecules) (Marcus, 2001). A theory of a neuronal physical symbol system must also explain the capacity to infer grammar during the process of language acquisition (Steels and Szathmáry, 2008). Finally, symbol grounding (i.e. semantic interpretability) is also needed.

We believe that strong evidence for a neuronal PSS comes from the field of grammar learning. The following is an example of a behavior that is proposed to require neuronal symbols. Gary Marcus has shown that 7 month old infants can distinguish between sound patterns of the form ABA vs. ABB, where A and B can consist of different sounds, e.g. “foo”, “baa” etc. Crucially, these children can generalize this discrimination capacity to new sounds that they have never heard before, as long as they are of the form ABA or ABB. Marcus claims that performance in this task requires that the child must extract “abstract algebra-like rules that represent relationships between placeholders (variables), such as “the first item X is the same as the third item Y”, or more generally that “item I is the same as item J”” (Marcus et al., 1999). Several attempts have been made to explain the performance of these children without a PSS (e.g. using connectionist models) (Seidenberg and Elman, 1999) but Marcus (2001, p. 70) has criticized these as smuggling in symbolic rules in one way or another by design. For Marcus it seems that the system *itself* must discover the general rule. In summary, the problem with a large set of connectionist learning devices is that a regularity learned in one component of the solution

representation is not applied/generalized effectively to another part (Marcus, 2001). Marcus calls this the problem of *training independence* (Marcus, 2001). Marcus considers this as one of the fundamental requirements for a learning system to be described as symbolic or rule based.

It is important to realize that it would be nonsense to claim that the brain is nothing but a physical symbol system. Indeed, we believe that a PSS is needed to explain only some relatively advanced aspects of cognition, e.g. some aspects of language and abstract thought. There is a huge amount of non-symbolic functionality possessed by neuronal processes. Non-symbolic learning mechanisms are probably utilized in the search for symbolic rules during ontogeny. An excellent example of such non-symbolic (connectionist) functionality is how the visual system can learn shift-invariance from a few training examples. The test for the trained system is to be able to determine whether two novel objects presented at different times or places are the same or different, e.g. for face recognition (Wiskott and Malsburg, 1995). Note that the same/different distinction is important in being able to solve Marcus’ ABA vs. ABB task. Konen and von der Malsburg (1993) have shown how shift-invariant pattern recognition can be achieved by rapid reversible synaptic plasticity (dynamic link matching). Exactly this process can be applied to automatically learning to distinguish same and different in a symbol system if the symbol is represented on a grid in the same way as a visual image. Thus, non-symbolic mechanisms can be involved in the discovery of symbol systems. The framework presented in this paper emphasizes this interaction. Also, we certainly do not claim that physical tokens are as simple as the 3×3 spatiotemporal structures we discuss. Instead, we propose that such tokens will be grounded in non-arbitrary ways to sensory and motor systems. Due to space constraints, the symbol grounding problem (semantic interpretability) is not addressed in this paper, although it is dealt with thoroughly elsewhere in a manner which does not remove the need for a physical symbol system (Harnad, 1990). Some authors have claimed that the symbol grounding problem has actually been solved in robotics (Steels, 2007). However, in these cases, a physical symbol system is still required.

To summarise, the following is a definition of a physical symbol system of the type proposed to be required to explain the kinds of rule learning exhibited in Marcus’s task above, adapted from Harnad (1990). A physical symbol system contains a set of arbitrary **atoms (or physical tokens)** that are manipulated on the basis of “**explicit rules**” that are likewise physical tokens or strings (or more complex structures consisting) of such physical tokens. The explicit rules of chemistry for example allow the calculation of reactions from the structure of atoms and molecules. The rule-governed symbol-token manipulation is based purely on the shape of the symbol tokens (not their “meaning”), i.e., it is **purely syntactic**, and consists of “rulefully combining” and recombining symbol tokens. There are primitive atomic symbol tokens and **composite symbol-token strings (molecules)**. The entire system and all its parts – the atomic tokens, the composite tokens, the syntactic manipulations both actual and possible and the rules – are all “**semantically interpretable**.” The syntax can be systematically assigned a meaning, e.g. as standing for objects, as describing states of affairs (Harnad, 1990). Semantic interpretability in molecular systems occurs in evolved biochemical systems, e.g. cell signaling molecules, transcription factors, etc., all convey meaning to a gene regulatory system about the state of the environment. Analogously, a neuronal symbol system contains information about the environment.

We will demonstrate how: (1) arbitrary physical tokens (atoms), (2) arranged into molecules or symbol structures,

(3) undergoing explicit rule-governed symbol-token manipulation (reactions), and (4) learning of explicit rule sets to produce functional symbol systems, are possible within a neuronal framework. The basic claims of this paper are as follows:

- (1) Atoms (symbol-tokens) have a dynamic existence as spatiotemporal patterns of spikes existing on neuronal chains resembling synfire chains, or more complex topologies of synaptic connectivity (Abeles, 1991; Doursat and Bienenstock, 2006; Ikegaya et al., 2004).
- (2) Formation of molecules is by ordering of spatiotemporal spike pattern symbol-tokens on a chain or between chains.
- (3) Explicit rules are implemented by context-sensitive re-write rules implemented by spiking neurons that read and write to neuronal chains.
- (4) Learning of an appropriate symbol system is done by a neuronally implemented learning classifier system in which spiking neuron classifier function is replicated using supervised learning by one spiking neuron of the input/output correlations of another spiking neuron.

The next section presents the details of the neuronal dynamics and plasticity model used. The following section demonstrates each of the four claims made above. Finally, the Discussion critically compares this framework with existing proposals for how a PSS can be neuronally implemented.

2. The spiking neural network and plasticity model

We use a neuronal and plasticity model, similar to that used in (Izhikevich, 2003, 2007). This models cortical spiking neurons with plasticity governed by spike-time-dependent plasticity (STDP) modulated by dopamine (DA) reward. In Izhikevich's work, the STDP function determines weight change *indirectly*, by producing a synaptic tag or eligibility trace molecule. It is the interaction between this eligibility trace at each synapse and the global reward signal (DA) that results in weight change. In our work, we typically set the decay of the eligibility trace to equal the decay of STDP, so that in effect there is no eligibility trace, but we do not exclude the possibility that a slower decaying eligibility trace could be useful in some learning situations. Furthermore, in some experiments the decay rate of STDP is assumed to be higher than in Izhikevich (2007), increasing the temporal-resolution of interspike-interval detection operations. The details of neuronal model are now presented.

2.1. Spiking neuronal model

Neurons were either excitatory neurons of the regular spiking type or inhibitory neurons (Izhikevich, 2006). The spiking model is from (Izhikevich, 2003):

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I, \quad (1)$$

$$\frac{du}{dt} = a(bv - u) \quad (2)$$

with resetting after a spike as follows.

$$\text{if } v \geq +30\text{mV, then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (3)$$

where v represents membrane potential, and u represents a membrane recovery variable. When v reaches +30 mV (the apex of the spike, not to be confused with the firing threshold), v and u are reset. For excitatory neurons $b=0.2$, $c=-65$, $a=0.02$, and $d=8$,

corresponding to cortical pyramidal neurons with regular spiking. For inhibitory neurons $b=0.25 \pm 0.05$, $c=-65$, $a=0.02 \pm 0.08$, $d=2$, and I is the input from other neurons, and external sources. External inputs to the network are given as 1 ms depolarizations of 30 mV. The details of the plasticity model are now presented.

2.2. STDP based plasticity model

As described in Izhikevich (2007) each synapse has two variables, a synaptic weight w and an eligibility trace e .

$$\frac{de}{dt} = -c/\tau_e + \text{STDP}(\tau)\delta(t-t_{pre/post}), \quad (4)$$

$$\frac{dw}{dt} = eD \quad (5)$$

where D is the extra-cellular concentration of DA in μM , and $\delta(t)$ is the Dirac delta function that increases e by an amount specified by the STDP rule. The STDP rule works as follows: τ is the time difference in milliseconds between pre- and post-synaptic spiking, and the eligibility of a synapse changes according to the standard implementation of additive STDP shown in Eq. (6) (Song et al., 2000; Izhikevich and Desai, 2003). The parameters A_+ and A_- effectively correspond to the maximum possible change in the synaptic weight per spike pair, while τ_+ and τ_- denote the time constants of exponential decay for potentiation and depression increments, respectively. In our experiments τ_+ and $\tau_- = 20$ ms (and sometimes 5 ms for high temporal resolution experiments), and $A_+ = 1.0$ and $A_- = 1.5$.

$$\begin{aligned} \text{STDP}(\tau) &= A_+ \left(1 - \frac{1}{\tau_+}\right)^\tau, \quad \text{for } \tau > 0 \\ \text{STDP}(\tau) &= A_- \left(1 - \frac{1}{\tau_-}\right)^{-\tau}, \quad \text{for } \tau \leq 0 \end{aligned} \quad (6)$$

Fig. 1 shows the STDP function.

One can see that if a pre-synaptic spike reaches the post-synaptic neuron (taking into account conduction delays) before the post-synaptic neuron fires, then $\text{STDP}(\tau)$ is positive. If a pre-synaptic spike reaches a post-synaptic neuron after it fires, then $\text{STDP}(\tau)$ is negative. The eligibility trace e decays with time constant $\tau_e = 20$ ms (or 5 ms, where specified to achieve higher temporal resolution). The synaptic weight w changes as the product of e and D . Weights were limited to a range from 0 to 30 mV. In the learning experiments, Dopamine reward ($D = 1.8 \mu\text{M}$) is provided at the same time as sub-threshold post-synaptic depolarization and exponentially

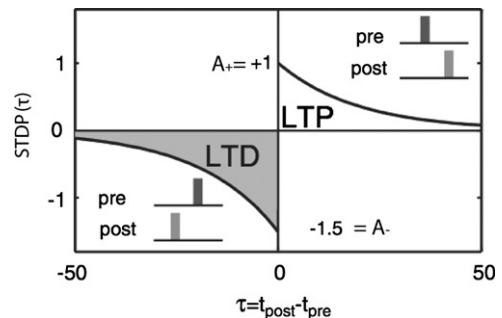


Fig. 1. Asymmetric profile of the STDP learning window. The difference between the timing of pre- and post-synaptic firing ($s = t_{post} - t_{pre}$) determines the degree and direction of the change in synaptic weight (Δw). The profile of the exponential potentiation ($s > 0$) and depression ($s < 0$) windows, respectively, are defined by the maximum possible weight change per spike pair (A_{\pm}) and the time constants of decay (τ_{\pm}). The ratio of integrals over these windows is defined as $\alpha = A_- \cdot \tau_- / A_+ \cdot \tau_+$ and thus a value of $\alpha > 1$ indicates an overall dominance of depression while $\alpha < 1$ indicates an overall dominance of potentiation. (From Izhikevich, 2007, Fig. 1).

decays with time constant 0.2 s. The implementation of STDP is as in Izhikevich (2006) except that weight updates are carried out every 1 ms according to Eq. (5).

In addition to standard inhibition, a longer time scale neuromodulatory inhibition is implemented. A neuromodulatory inhibitory neuron may release a substance when it spikes, reducing the sensitivity in the afferent neuron by a multiplicative term. This term is normally 1, but is set to 0 when a spike from a neuromodulatory neuron is received. It linearly returns to 1 by an increment of 0.004 per millisecond. This is important in allowing gating that is not critical on exact inhibitory spike timing.

3. Implementation of atomic symbols

We propose that an atomic neuronal symbol-token has the form shown in Fig. 2.

The top of Fig. 2 shows four examples of symbol-tokens consisting of spatiotemporal patterns of spikes. The y-axis indicates which neuron the spike will stimulate, and the x-axis indicates time. Thus, the depiction of the (purple) spatiotemporal pattern on the left indicates that the middle neuron is stimulated 10 ms later than the top and bottom neurons. The remainder of the figure shows the consequences of stimulating a chain of neuronal connections with this spike pattern. Each chain consists of three synapses, e.g. these may be cortico-cortical or cortico-thelamic connections with axonal delays of up to 50 ms at each step. Assume that all the delays are equal in this system (although this is not a critical requirement for the model). After the first three neurons (on the left of the chain) are stimulated with the spatiotemporal spike pattern, they will be asynchronously activated. Spikes will pass down the chain, asynchronously activating the second and third layers. The input spatiotemporal pattern can be reconstructed at many points along the chain. It is this spatiotemporal pattern of spikes that we define as an atomic neuronal symbol-token. The diagram shows that detector neurons at various locations along the chain can detect this spatiotemporal spike pattern if the axonal delays from the pre-synaptic neuron to the detector neuron are appropriately configured, e.g. if a spike from the top and bottom neurons take 11 ms to reach the detector,

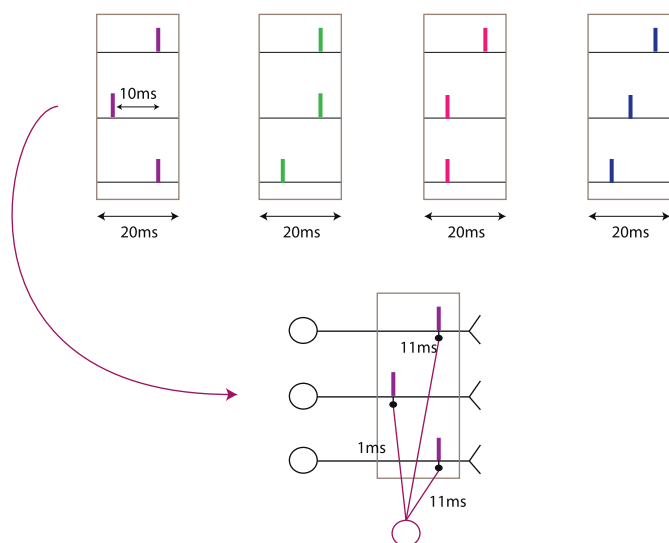


Fig. 2. Four possible spatiotemporal spike pattern based symbol-tokens are shown at the top. Below we see how the symbol passes down a delay line consisting of 3 neurons from left to right. The detector neuron (purple) can detect the spike pattern in several locations along the chain. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

but a spike from the middle neuron only takes 1 ms to reach the detector. If a voltage contribution from each neuron is necessary to fire the detector, then the detector will fire only when the appropriate spike pattern is seen. This implementation of neuronal symbol-tokens (atoms) uses the concept of polychronous computing and a modification of the concept of wavefront computing (Izhikevich et al., 2004; Izhikevich and Hoppensteadt, 2009).

Fig. 3 shows a simulation of a simple inter-spike interval detector of the type described above. The circuit shown consists of two input neurons with axonal delays of 10 and 1 ms into a post-synaptic neuron. The voltages are set so that two simultaneously arriving spikes are needed to trigger the output neuron. The graph below shows the neuronal voltages measured in an experiment in which the two input neurons were stimulated by external current to produce spikes over the course of a two second experiment. The first neuron is triggered at 100 ms intervals and the second neuron is triggered at 101 ms intervals. The input pattern is shown in Fig. 2. Red and green lines show input voltages and the thick blue line shows the output voltage. Because of the different stimulation periods of the inputs, the inter-pulse interval progresses from +1 to +20 ms over the course of the 2 s experiment.

It is observed that only for inter-spike intervals of 8 and 9 ms does the post-synaptic neuron (dark blue) fire. Note that if the weights are further decreased, the temporal specificity of the post-synaptic neuron increases. Also note that the refractory period of the neuron imposes a strong constraint with respect to the number of times a neuron can fire in one second and hence to the number of inter-spike interval calculations that can be made per second by a post-synaptic neuron. This limits the temporal resolution and complexity of spatiotemporal spike patterns that can be detected. If there is a time limit on the temporal duration of a symbol, e.g. dictated by the wavelength of a brain rhythm, then this temporal resolution imposes a constraint on the alphabet size of symbol-tokens.

4. Implementation of molecular symbol structures

The construction of molecular symbol structures from atomic symbol-tokens requires **binding** of atomic symbol-tokens together (Biederman, 1987; Malsburg, 1999) such that they can be subsequently manipulated (reacted) as a function of the structure of the molecule. In our framework, compositional neuronal symbolic structures exist as temporally ordered sequences of symbols along chains of neurons, see Fig. 4.

Fig. 4 shows a snapshot of the state of a neuronal chain that carries the four symbol-tokens shown in Fig. 2. Imagine producing this pattern by stimulating the first three neurons on the left with the blue, purple, green and finally pink spike patterns in succession. Let us allocate each spatiotemporal pattern an arbitrary label, e.g. pink=A, green=B, purple=C, and blue=D for convenience. Then this symbol-structure can be described as a string or linear molecule of the form **ABCD**. Many such short chains can exist in the brain. Each chain can be considered to be a kind of register, blackboard or tape that can store any symbol-tokens of the appropriate size. Figs. 2 and 3 show how a single symbol-token (C) could be read by a detector neuron with the appropriate axonal delay pattern when interfacing with the chain. Similar detector neurons can exist for the symbol-tokens A, B and D and as many others as spatial width of the chain and temporal resolution allow.

Thus, we envisage a potentially large parallel system in the brain consisting of a *population* of such chains (chains in the sense of topological and not necessarily spatial organization), each capable of storing a set of symbol-token strings and operating

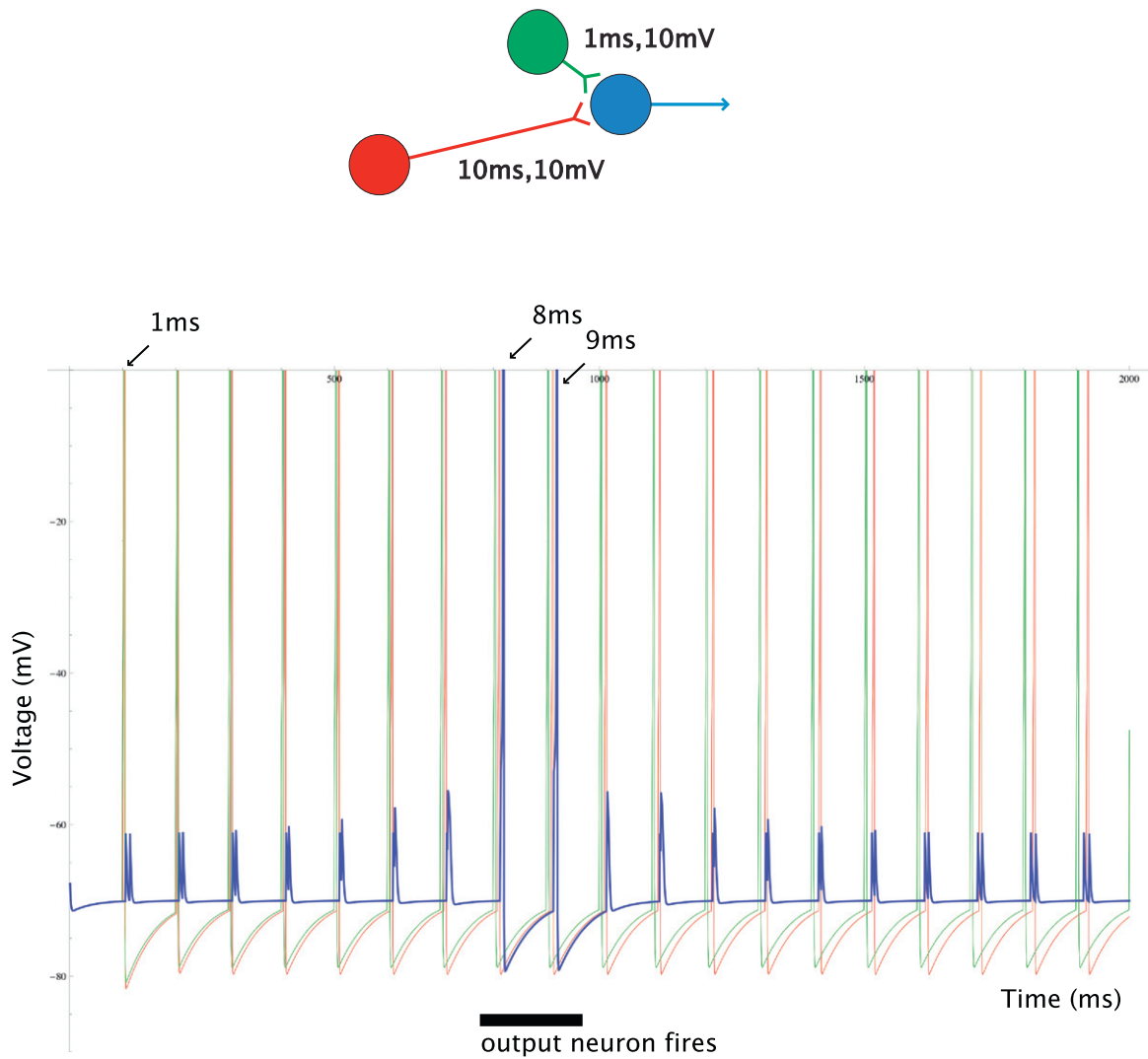


Fig. 3. (Top) A simple classifier neuron sensitive to a spike pattern occurring 10 ms apart. Weights = 10 mV. The experiment involves changing the inter-spike interval from 0 to 20 ms.



Fig. 4. Chain carrying 3 different spike patterns as a concatenated string.

on these strings in parallel. Interaction between (and within) such chains constitutes the operations of symbol-manipulation. Returning to the chemical metaphor such interactions can be thought of as chemical reactions between molecules contained on separate chains, and rearrangements within a molecule expressed on the same chain. Whilst in a sense a chain can be thought of as a tape in a Turing machine (due to the serial nature of the strings), it also can be thought of as a single molecule in a chemical system (due to the existence of multiple parallel chains). This constitutes the core representational substrate on which symbol

manipulation will act. The reactivity of symbol structures on these chains is described in the next section.

5. Implementation of reactivity rules

5.1. Implementation of implicit rules

A fundamental operation that can be carried out upon a symbol-token is to replace it with another symbol-token, see

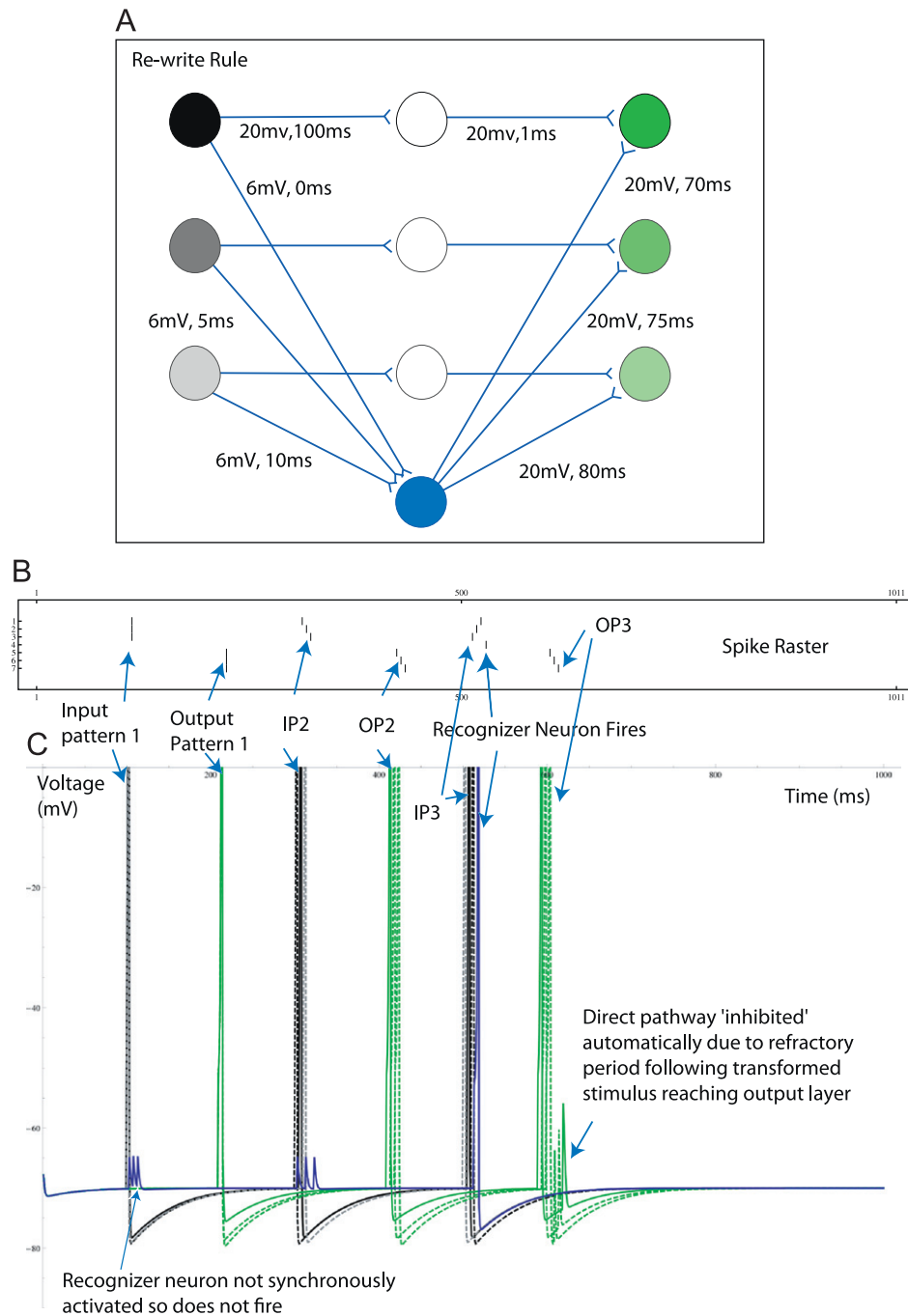


Fig. 5. (A) Re-write rule is implemented by the above circuit. There are three input channels in this case, although it is trivial to add more. The direct pathway is by a delay line via an intermediate layer. The indirect pathway to the outputs is via a classifier neuron (blue). Only if the delays match the ISIs of the input spike ensemble does the recognizer fire. Once fired, it sends signals down outputs with delays that are set so that the desired output pattern is produced. (B) A spike raster showing the 3 input patterns and 3 output patterns produced in an experiment. Patterns that do not match the re-write rule pass through the classifier, but those that do match the re-write rule are converted, and the passage through by the original pattern is inhibited due to the refractory period of the output neurons (see (C) which shows the voltages of input, output and classifier neuron). Also it is possible to explicitly inhibit the passage of the original input, but this is not needed here. Black=3 inputs, green=outputs, blue=classifier neuron. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 5. The network figure shows a chain again of three neurons wide. Receiving input from the chain and writing activity back into the chain is a detector neuron with specific input and output delays in relation to the chain. A detector neuron (blue) only fires when the correct pattern of input is detected (as described above). In this case, the neuron's input delays are set so that it recognizes (fires for) patterns only of type *D*.

In the experiment the following pattern of stimulation was given. The spike raster plot and the voltage plot show two

spatiotemporal patterns input to the input neurons that fail to make the classifier neuron fire. It can be seen that the same pattern enters the chain as leaves the chain. This is because the spatiotemporal organization of these patterns does not match the tuning curve of the detector neuron. Only when the third spatiotemporal spike pattern is input does the detector neuron fire. Once fired, the output of the detector neuron is sent back to the neurons of the chain. If the output of the detector neuron slightly precedes the normal passage of the untransformed

pattern through the chain, then the refractory period of the output neurons of the chain prevents interference by the original untransformed pattern which is thereby replaced by the new pattern specified by the detector neuron. Such a detector neuron we will now call a classifier neuron because it is a simple context free re-write rule with a condition (detection) and an action pole of the type seen in Learning Classifier Systems (LCS) (Holland and Reitman, 1977).

It can be seen that such classifier neurons are selective filters, i.e. the classifier neuron is only activated if the spatiotemporal pattern is sufficiently matched with the axonal delays afferent upon the neuron, for the neuron to fire. The above classifier implements an implicit rule. An implicit rule is a rule that operates on atomic or molecular symbol structures without being specified (encoded/determined/controlled) by a symbol structure itself. There is no way that a change in the symbol system, i.e. the set of symbols in the population of chains, could modify this implicit rule. The implicit rule is specified external to the symbol system. Whenever the symbol D passes along this chain, it will be replaced by the new symbol, irrespective of the presence of other symbols in the system.

5.2. Implementation of explicit rules

In a symbol system (as in chemistry), symbols are manipulated (partly) on the basis of “**explicit rules**”.¹ This means that the operations or reactivity of symbols depends on/is controlled by/is causally influenced by their syntactic and semantic relationship to other symbols within the symbol-structure and between symbol structures. Fig. 5 shows a classifier neuron implementing an implicit rule. This rule was not controlled by any symbols in the system, it merely operated on symbols in the system. Fig. 6 shows that a classifier neuron and an inhibitory gating neuron can implement an explicit rule within our framework.

The classifier and chain shown in Fig. 5 is simply modified to include an inhibitory gating unit that must receive a particular pattern of spikes (T for trigger) in order for it to become active. The simplest relation is where T immediately precedes X. Only when this is the case will the classifier neuron be disinhibited. Only when the classifier neuron is disinhibited will X be converted to Y. Otherwise X will pass through an inactive classifier (as will all other symbols). This is formally a context-sensitive rewrite rule. The rule is called context sensitive because the conversion of X into Y depends on the relation of X to another symbol T. The semantic meaning of T in this syntactic relation to X is to cause the reaction of X to Y. A set of context-sensitive rewrite rules is capable of generating a grammar of spike-patterns. Consider starting the system off with a single symbol-token S. Probabilistic application of the rules to the initial symbol S would result in the systematic production of spike patterns consisting of grammatically correct context-sensitive spike pattern based sentences.

¹ Quoting Harnad (1990, p. 335). “Wittgenstein (1953) emphasized the difference between explicit and implicit rules: It is not the same thing to “follow” a rule (explicitly) and merely to behave “in accordance with” a rule (implicitly). The critical difference [between an implicit and explicit rule] is in the **compositeness** (7) and **systematicity** (8) criteria. The explicitly represented symbolic rule is part of a formal system, it is decomposable (unless primitive), its application and manipulation is purely formal (syntactic, shape-dependent), and the entire system must be semantically interpretable, not just the chunk in question. An isolated (“modular”) chunk cannot be symbolic; being symbolic is a systematic property... For systematicity it must be possible to combine and recombine entities rulefully into propositions that can be semantically interpreted... It is possible to devise machines whose function is the transformation of symbols, and whose operation are sensitive to the syntactical structure of the symbols that they operate upon.”

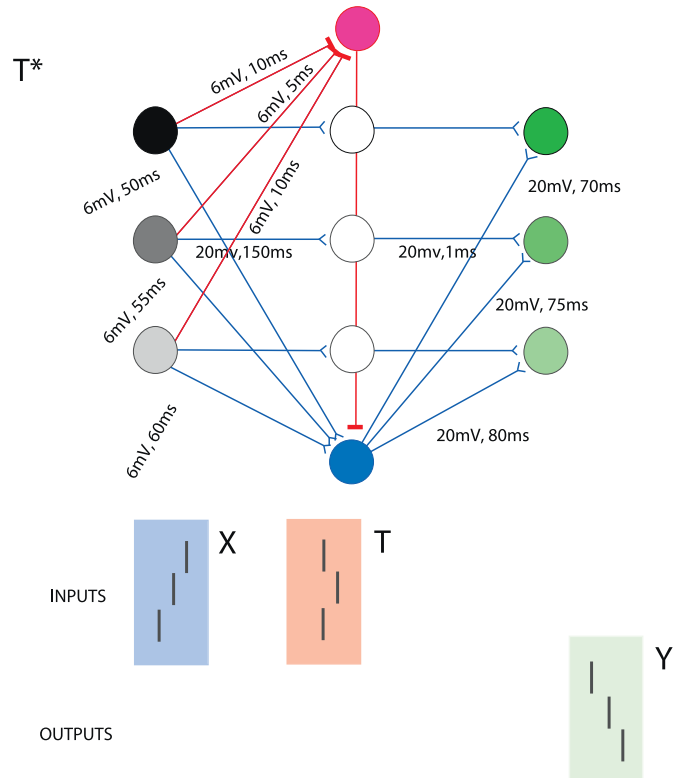


Fig. 6. An explicit rule implemented by a classifier neuron and an inhibitory gating neuron. The inhibitory connection from the inhibitory neuron to the classifier neuron is shown in orange. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

However, the system so far described could not easily implement the kind of rule that Marcus wishes a symbol-manipulation system to learn, namely to extract “abstract algebra-like rules that represent relationships between placeholders (variables), such as “the first item X is the same as the third item Y”, or more generally that “item I is the same as item J”” (Marcus et al., 1998). This kind of rule requires hash symbols which implement the concept of same and different, namely, If #₁ # #₁ then S, Else If #₂ # #₁ then D. That is, if the first and last string are the same, write S=same, and if the first and last strings are different write D=different. In the absence of hash symbols of this type, a classifier system would have to learn all the explicit rules for each possible pair of symbols at the first and last position, instead of learning the general rule. Both systems would be systematic; however, the system with hashes would allow a more concise specification of the same level of systematicity. But how can such hashes be implemented within our framework?

Fig. 7 shows one possible method of detecting when two spatiotemporal spike patterns are the same or different.

On the left, the figure shows two pairs of sequentially presented symbols flowing down two reaction chains, in this case, AA on the top chain and AB on the bottom chain. On the right we see that the symbols AA from the top chain have been sent to a chain that is capable of recognizing same/different. This circuit is very simple and consist only of three XOR gates implemented by spiking neurons. The XOR function is at the heart of same/different classification because it fires 1 for the inputs 01 and 10, but fires 0 for the inputs 00 and 11. In this case, if two spikes are separated by 100 ms along each channel then they will cancel each other out. However, if only one spike is present then it will be capable of activating the XOR gate. By setting the threshold of the output neuron it is possible to detect adjacent symbol tokens that differ by some specified number of

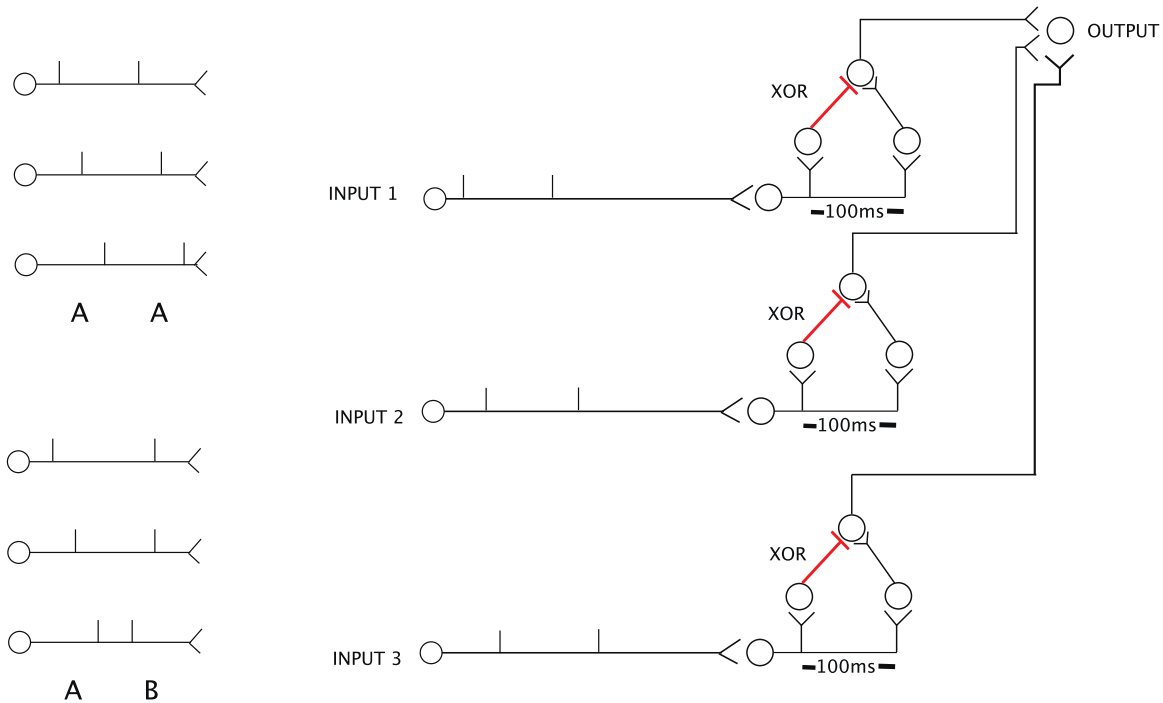


Fig. 7. A method for detecting same/different successive symbol-tokens. (Left) Two pairs of sequentially presented symbols, AA and AB. (Right) A device that is capable of identifying symbol pairs that are different, using three XOR circuits in parallel. The circuit on the left has been stimulated with AA. Each of the 100 ms inter-spike interval (ISI) detectors will fire in this case. The summed depolarization of these three detector neurons will accumulate to fire, and thus signal that two identical symbols separated by a 100 ms was observed. If AB had been presented then only the middle 100 ms ISI detector neuron would have fired, and this would not be sufficient to fire the output neuron.

spikes. The output neuron can write to the channel in the same way as described for the implicit rule action, e.g. implementing the rule, If #₁ # #₁ then S.

Note that for the above detector to be generally applicable to any two symbols that are the same, it is necessary that the output neuron accumulate depolarization from the three detector neurons over at least 100 ms. If this was not the case then the axons from the detectors to the output neuron would have to have very specific time delays that would only allow detection of a specific pair of same symbols, e.g. AA but not BB.

It seems that the neuronal capacity for detection of same and different is a significant departure from what can easily be achieved in chemistry. A neuronal physical symbol system is capable of exploiting generalization mechanisms unavailable to chemistry. In chemistry there is no known molecular mechanism by which one molecule can determine whether two other molecules are the same or different.

This concludes our presentation of the structures and reactions possible in a neuronal symbol system using spiking neuronal networks. We now address the more difficult question of how a symbol system can be learned. We first outline the symbol learning system at the algorithmic level, and then demonstrate how one of the fundamental operations of this system, namely “rule evolution” can be implemented by replication of classifiers using supervised learning.

6. Neuronal evolutionary algorithm for learning a symbol system

There have been several demonstrations of the ability to discover a functional symbol system on the basis of reinforcement learning. A powerful architecture is XCS (accuracy based classifier system) shown in Fig. 8 (Butz, 2006).

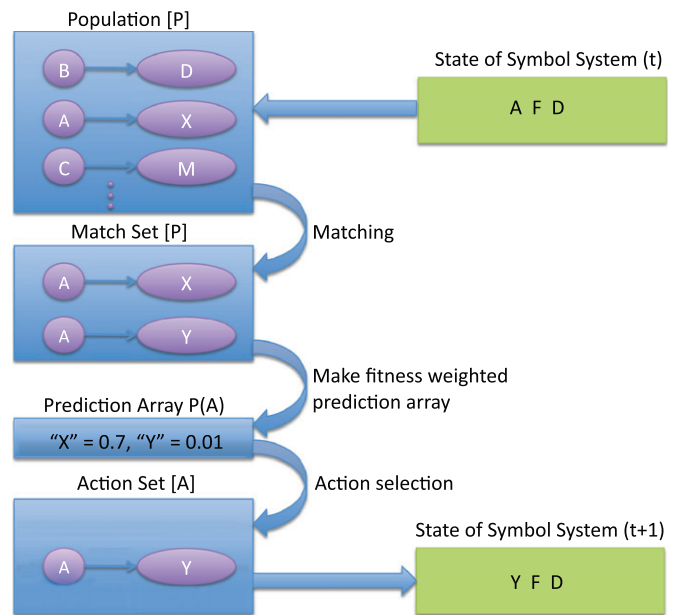


Fig. 8. An example of an XCS classifier system. The symbol system has state AFD at time t . Of the population of classifier neurons only those with A as their condition can match. Of these the classifier with highest predicted reward is that which transforms A to Y. This is applied. Classifier fitness is on the basis of reward prediction accuracy. A GA is undertaken on the Match Set.

The XCS algorithm consists of a population of classifiers with condition-action poles, $C \rightarrow A$. Each classifier has a fitness F that is related to its accuracy in predicting the reward obtained in the next time step. At each point in time a subset of the classifiers (called the Match Set) will match the state of the environment.

Classifiers proposing several possible actions may exist in the Match Set. An action selection method is used to select the best classifier most of the time, although sometimes actions using sub-optimal classifiers are also executed for the sake of exploration. When the action is executed and the reward obtained, then the prediction accuracy of the classifiers in the action set can be updated. Selection then takes place between classifiers in the Match Set, while those with lower fitness are removed from the population. This is effectively a niche-based selection that preserves representational diversity in the population of

classifiers. Learning classifier systems have been used to evolve classifiers for reinforcement learning tasks such as navigation, robotic control, but also for function approximation (Bull and Kovacs, 2005).

We propose that a learning classifier system is implemented within the brain and is necessary for discovery of symbol systems capable of solving problems like ABA vs. ABB discrimination. The capacity to use a learning classifier system for such grammar learning tasks has been investigated elsewhere (Cyre, 2002; Unold, 2008). In other words, we propose that a kind of highly

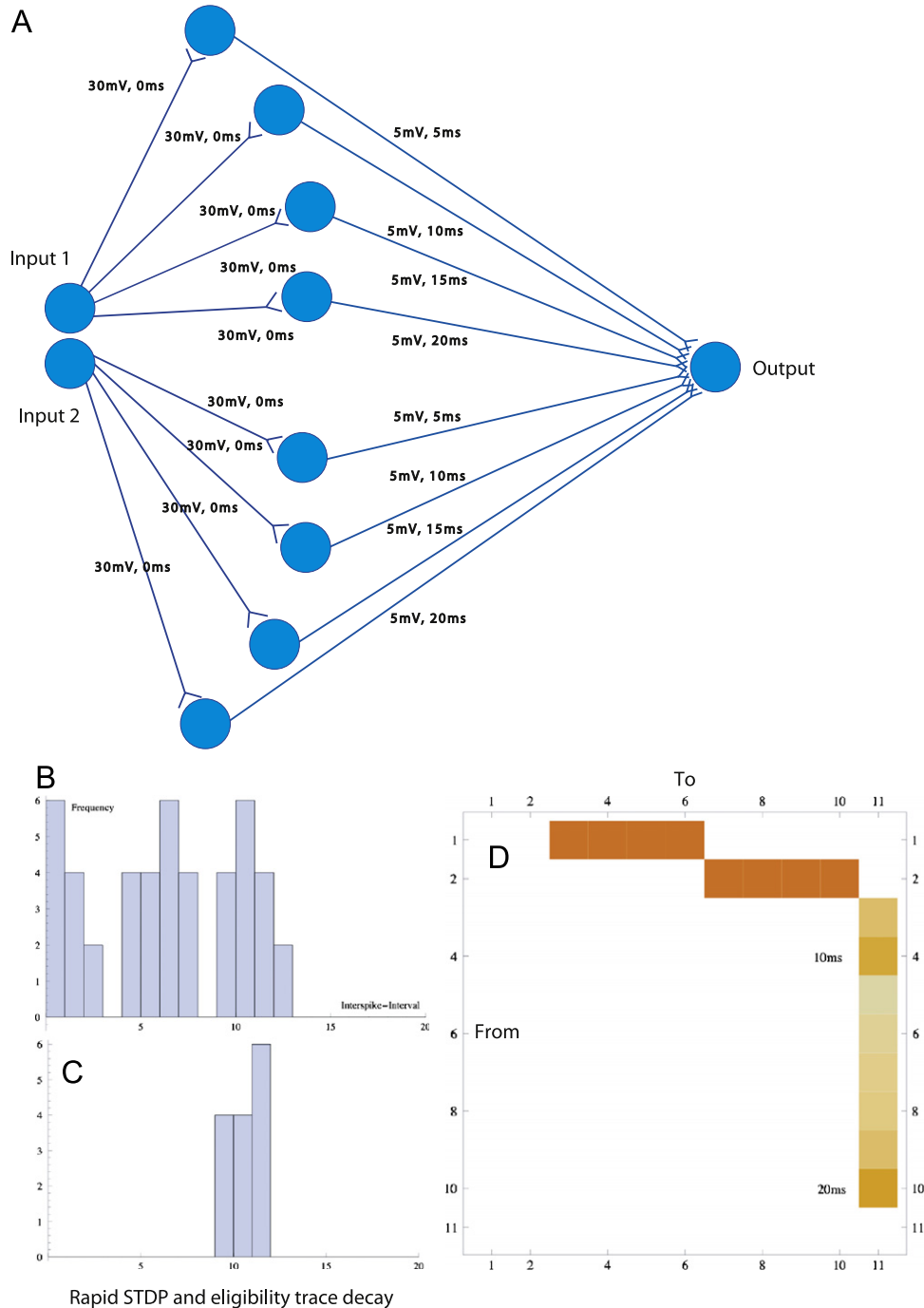


Fig. 9. (A) Architecture of the device to be trained to detect ISI=10 ms. (B) Giving reward and sub-threshold depolarization (5 mV) at 20 ms after the first spike of the pair, and using plasticity rules described previously (Izhikevich, 2007), we found that training was non-specific for 10 ms ISI. This is a histogram of the number of times the output neuron fires for various ISIs during the test phase (i.e. with no post-synaptic depolarization given.) (C and D) However, after increasing the rate of decay of STDP and eligibility trace, training was specific. The selected delay from input 1 to output 1 was 10 ms and the selected delay from input 2 to output 2 was 20 ms, providing a good response to a 10 ms ISI at 20 ms after the first spike, as we intended.

constrained natural selection takes place in the brain to evolve classifiers. This is a part of a larger hypothesis called the Neuronal Replicator Hypothesis which proposes that replicators exist in the brain and can undergo natural selection (Fernando et al., 2010, 2008; Fernando and Szathmáry (2009a,b); Szathmáry and Fernando, 2009).

In order for the argument that an LCS is implemented in the brain to be plausible, it is necessary to explain how it is possible to replicate classifiers of the type shown in Figs. 5 (implicit) and 6 (explicit). This question is answered in the following section.

7. Replication of neuronal classifiers

There are several steps to obtain replication of classifiers. The first is to understand how a single classifier can be trained.

Using the STDP based synaptic plasticity rules described previously it is possible to train a classifier neuron to fire only when exposed to a particular spatio-temporal pattern of spikes, as shown in Fig. 9.

In this experiment we wished to train the output neuron to fire only for a particular interspike interval between the two input neurons. In other words, we wanted to train a classifier neuron to have the same functionality as Fig. 2. The architecture of the circuit is as follows. We assume that each input neuron has many pathways for communicating with the output neuron. For example dendrites form the post-synaptic neuron may connect with the axon of the pre-synaptic neuron at many locations. Alternatively, it may be the case that several neurons are involved in the path from input to output neuron. In the model we assume that strong pathways of insignificant delay exist from the input neuron to a set of nearby neurons, and that from each of these nearby neurons there is a pathway to the output neuron with delays of 5, 10, 15, and 20 ms each, see Fig. 9. Each weight from these intermediate neurons to the output neuron is initially sub-threshold (5 mV). This means that 3 intermediate neurons must fire for the output neuron to fire. Because only two pre-synaptic neurons can contribute to a synchronous pulse, the output neuron should therefore never fire! Indeed, only if a sub-threshold depolarization is provided by an external teacher to the output neuron, will it fire, if at that same time it is sufficiently stimulated by pre-synaptic neurons.

In the experiments, sub-threshold (training) depolarization (10 mV) of the post-synaptic output neuron was given 20 ms after the desired condition spike pattern was presented to the input neurons. Due to STDP the weights from the intermediate neurons to the output neuron increased. Training was with spike pairs every 100 ms for 2 s. ISI is either 0, 5, 10, 15 or 20 ms. Only when the ISI was 10 ms was (Dopamine) reward given at 20 ms after the first input spike. Fig. 9B shows that even though reward and post-synaptic sub-threshold depolarization is given only 20 ms after a 10 ms inter-spike interval, there is still the reinforcement of weights that result in sensitivity to inter-spike intervals of 0, 5 and 10 ms. In other words, the temporal sensitivity of training was too low. One reason was that the eligibility trace was too slow to decay in comparison with the spike pairs that were being presented every 100 ms. Increased specificity is obtained by the following modifications. STDP and eligibility traces adjusted to decay by 40% every millisecond rather than 5% every millisecond. A reward bolus of 0.36 units is given at the same time as the 10 mV post-synaptic depolarization at 15 ms. The performance obtained is shown in the tuning curve of the output neuron in Fig. 9C which shows the frequency of firing of the post-synaptic neuron (y -axis) against the interspike interval (x -axis) The 10 ms connection from input 1 and the 20 ms connection from input 2 have been reinforced to approximately

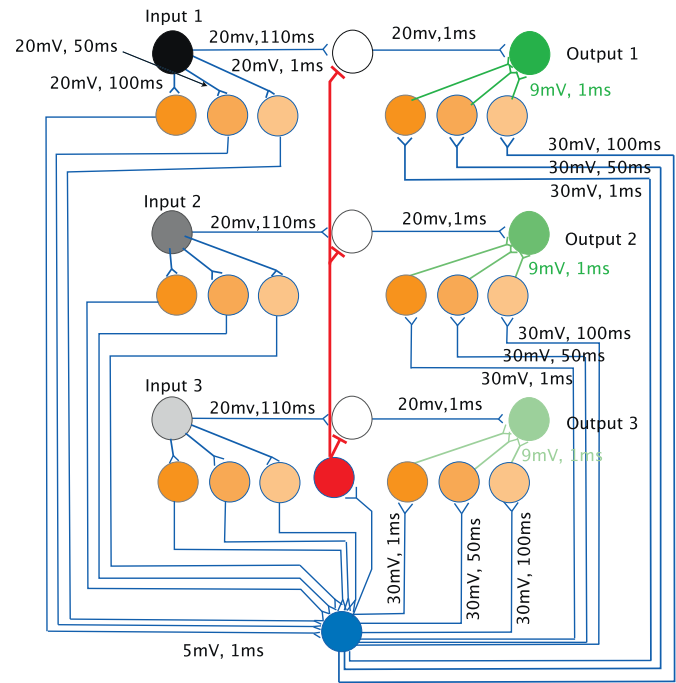


Fig. 10. The learning circuit capable of learning a grammatical re-write rule by supervision. Three input neurons (black and gray) receive the input symbol encoded as a 3×3 spatiotemporal matrix of spikes. That means it is distributed over 3 neurons, and there are 3 temporal positions a spike can take, $-1, 0, 1$ for example. These send signals to three neurons along delay lines of 100, 50 and 1 ms. Weak weights then connect these to a classifier neuron. Sub-threshold activation of the classifier neuron at 100 ms results in STDP causing potentiation of its afferent weights. Simultaneously, appropriately timed supra-threshold depolarization of the output neurons (green) causes potentiation of their afferent weights. The classifier neuron inhibits the direct pathway from input to output if the appropriate symbol is recognized. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

8.5 mV, see Fig. 9D for the weight matrix. Input neurons=1,2, intermediate neurons=3–10, output neuron=11. Thus, we showed that it is possible to train a classifier neuron to recognize particular interspike intervals.

The second step is to train a classifier capable of reading and writing a spatiotemporal spike pattern, see Fig. 10.

Fig. 10 shows again the usual chain of three channels in width consisting of three input neurons (gray) projecting to 3 output neurons (green). The training regime is as follows and is shown in its entirety in Fig. 11.

During the training period the spike pattern to be recognized entered along the 3 input channels with spikes at 0, 50 and 100 ms latency. This pattern was presented 9 times. A short fixed time period after each input pattern was presented to the input neurons, a pattern of supra-threshold depolarization was presented to the output neurons. This output pattern was the desired output pattern, which in this case is an inversion of the original pattern (although any pattern can be trained). Fig. 10 shows there is a set of alternative possible delay lines from the each input neuron to the blue classifier neuron, and another alternative possible set of delay lines from the classifier neuron to each output neuron. In addition, the classifier neuron is linked to a neuromodulatory inhibitory system that serves to block the passage of the original spike-pattern if it is recognized. If it is not recognized then the original pattern passes through to the outputs with a delay of 120 ms, unchanged in form.

To see the training regime in detail, Fig. 11 shows that an input pattern activates the input neurons 9 times followed by the

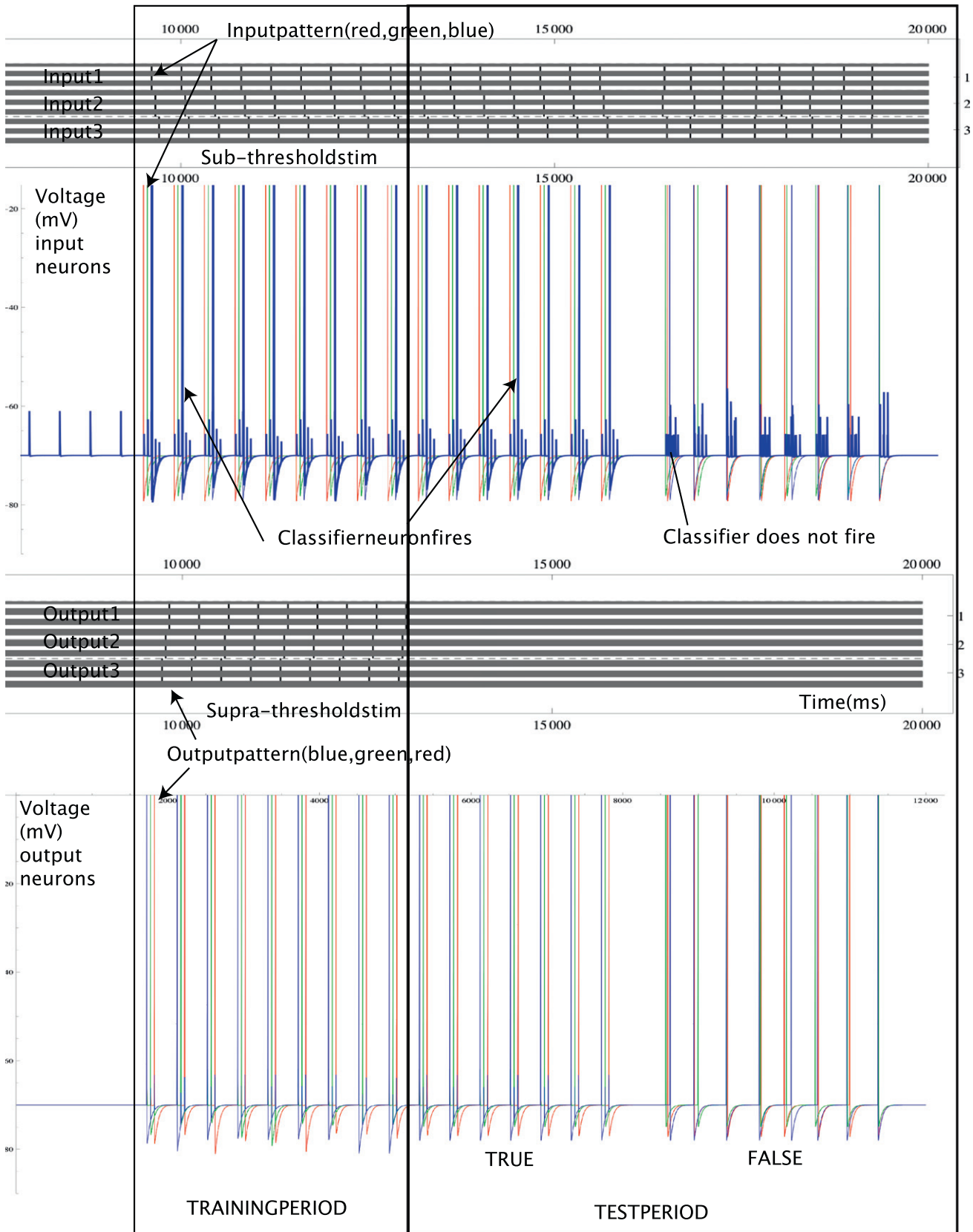


Fig. 11. Training and test phases during the experiment to construct the classifier shown in Fig. 10. See text for details.

desired output pattern. Output stimulation is supra-threshold (20 mV) and is provided at 218, 168 and 118 ms after the first input spike. In addition, the classifier neuron is given sub-threshold

stimulation (10 mV) 100 ms after the first input spike. This training procedure is sufficient for the classifier neuron to learn both the input required to activate it, and the desired output.

The test phase to prove that training has been successful is shown in two parts. In the part (true) the correct input pattern is given and the classifier is able to transform it into the desired output pattern. In the next phase (false) random inputs are given and pass directly through the chain unchanged because the classifier neuron is not activated by them.

The final step for classifier replication is identical to the training regime used to train the classifier in Fig. 11. This is because once a single classifier neuron has been trained, this classifier neuron can train *other* classifier neurons in the following manner. The plasticity of the first (trained) classifier neuron is held fixed. The input spike pattern passes now to both classifiers, and the output of the first classifier is used to produce supra-threshold output neuron depolarization in the second classifier.

Systems that are capable of being trained by supervised learning, are typically also capable of training other such systems.

8. Discussion

There are several alternative connectionist type theories for the implementation of 'mental representations' or symbol-structures in the brain. For example the RAAM architectures of Pollack (1990) encoded syntactically structured representations of sentences, and uses back-propagation in the space of representations with no explicit concatenative structure. However, this system cannot generalize in the way required by Marcus to new inputs. Also back-propagation is an unlikely mechanism in neocortex. Our framework on the other hand uses realistic plasticity rules (STDP and gating) and implements a real physical symbol system.

An alternative proposal for structured representations are semantic networks (Marcus, 2001). They are graphs with labeled connections. A serious problem with these systems is the speed limitation in creating new bindings. Bindings in our system are dynamic, arising from the spatiotemporal arrangement of patterns of spikes. Thus, the problem of new rapid binding formation does not exist.

An influential alternative theory of neuronal symbol systems is binding-by-synchrony. This theory proposes relational (bidirectional) binding of symbol-tokens. A symbol-token in this case is not the same as proposed here. Typically the activation of a particular node (neuron or group of neurons) is a symbol. Relational binding of such symbol-tokens is achieved by synchronous (phase locked) firing of a relation representing neuron with an instance representing neuron (Bienenstock and Geman, 1995; Shastri and Ajanagadde, 1993), for example, the subject and object rules of a verb, and the subjects and object instances. However, there are several problems with binding by synchrony.

Binding by synchrony is limited in the kinds of relation that can be made, e.g. the relation $A \rightarrow B$ is not trivial to distinguish from the relation $B \rightarrow A$ (Love, 1999) because binding by synchrony is symmetric. Therefore, directed graphs cannot naturally be represented using binding by synchrony. A directed graph is an important cognitive symbol structure (molecule) because it can represent cognitive and semantic graphs. Our framework allows the representation of directed graphs, e.g. a string can be a bracketed expression describing a graph, or it can be the output of a breadth first search of a directed graph.

The inference (reaction) rules that transform a symbol-structure into another symbol-structure are implicit not explicit, i.e. a particular rule linking predicates is not represented in the same way as the predicate itself. In terms of our chemical metaphor, the reactivity of a molecule is not dependent on the structure of that molecule, but on reactivity defined independently of the molecular structures. Our framework on the other hand implements explicit rules.

There is no mechanism for recursive binding because representation of multiple instances of a predicate results in a superposition catastrophe. (Marcus, 2001, p. 101). Also it is difficult to represent multiple instances of an object. Recursive binding and multiple instances can be represented in our system because it is a spiking neuronal network implementation of a real physical symbol system.

Novel predicates and rules can be learned rapidly in human communication, yet this would appear to require producing a new node in binding by synchrony models. How compositional representations are learned and modified, e.g. during language acquisition is not explained by Shastri et al. (Bienenstock and Geman, 1995). Our framework on the other hand allows rapid new binding of symbols by novel ordered concatenation of symbols on the same chain.

Finally, there is some neurophysiological evidence that may be in support of this framework for symbol processing. The discovery of "cortical songs" is highly suggestive that discrete unique tokens such as symbols can be encoded as spatiotemporal patterns of spikes. Cortical songs are higher-order sequences of spike patterns repeated in the *same sequential order* observed in neocortical brain slices, of the form $[A,C,D,E,F][A,C,D,E,F]$ for example where each letter represents a stereotyped polychronous pattern of activity (Ikegaya et al, 2004). Furthermore, there is evidence for the training methods we used to train classifiers in Figs. 9 and 10. For example, synaptic inputs at distal dendrites can act as supervisory signals in the Hippocampus (Dudman et al., 2007). This maps to the sub-threshold and supra-threshold depolarization we used to train classifier and output neurons. Several other papers also propose methods for supervised training of classifiers, and so our classifier replication mechanism is by no means out of the blue. For example, the "Tempotron" is an example of learning to classify specific spatiotemporal patterns of spikes using a gradient-descent type rule to adjust weights on the basis of how rapidly a pattern results in firing of a classifier leaky-integrator neuron (Gutig and Sompolinsky, 2006). A spiking analogy of the Widrow–Hoff algorithm has been developed by Ponulak and Kasinski for the supervised training of spiking neural networks to undertake sequence learning, classification and spike-shifting. It works by potentiating the weight from input to output whenever a target spike is observed and depressing it whenever the trained neuron fires. The extent of the effect is a decaying time function from the time of the last input spike at this synapse. Inhibitory synapses are driven by the opposite rule. When the desired and actual spike times are identical, potentiation and depression cancel (Ponulak and Kasinski, 2009). Therefore, there is a growing body of work showing how replication of spatiotemporal spike pattern classifiers is possible.

In conclusion we have attempted to show how in principle a physical symbol system capable of learning symbolic rules could be realized in the brain using spiking neuronal networks and replication of classifiers using STDP based supervised training/learning of functions, within a learning classifier framework. Our emphasis has been on the implementation aspects and not on the algorithmic aspects. Further work is needed to implement a full LCS in a neuronal network. Therefore, this paper should be seen as perhaps a qualitatively new way to think about symbol processing in the brain.

Acknowledgements

Many thanks to two anonymous reviewers for helping to considerably improve this manuscript. Also, thanks to Eors Szathmary, Phil Husbands, Daniel Bush, Luc Steels, Martin Butz, Anil Seth, Eugene Izhikevich, and Larry Bull for helpful discussions

and throughout our exploratory work on the Neuronal Replicator Hypothesis. Thanks to the FP7 E-FLUX EU grant and for a Marie Curie Fellowship to work at Collegium Budapest in Hungary.

References

- Abeles, M., 1991. *Corticomics: Neural Circuits of the Cerebral Cortex*. Cambridge University Press, New-York.
- Biederman, I., 1987. Recognition-by-components: a theory of human image understanding. *Psychological Review* 94 (2), 115–147.
- Bienenstock, E., Geman, S., 1995. Compositionality in Neural Systems The Handbook of Brain Theory and Neural Networks: MIT/Bradford Books. Elsevier.
- Bull, L., Kovacs, T. (Eds.), 2005. *Foundations of Learning Classifier Systems*. Springer, Heidelberg.
- Butz, M., 2006. *Rule-Based Evolutionary Online Learning Systems: A Principled Approach to LCS Analysis and Design*, vol. 191. Springer, Heidelberg.
- Clark, A., 1991. In Defense of Explicit Rules. In: William Ramsey, S.P.S.D.R. (Ed.), *Philosophy and Connectionist Theory*. Lawrence Erlbaum.
- Cyre, W., 2002. Learning grammars with a modified classifier system. In: *Proceedings of the Evolutionary Computation (CEC)*.
- Doursat, R., Bienenstock, E., 2006. Neocortical self-structuration as a basis for learning. In: *Proceedings of the Fifth International Conference on Development and Learning (ICDL 2006)*.
- Dudman, J.T., Tsay, D., Siegelbaum, S.A., 2007. A role for synaptic inputs at distal dendrites: instructive signals for hippocampal long-term plasticity. *Neuron* 56, 866–879.
- Fernando, C., Goldstein, R., Szathmáry, E., 2010. The neuronal replicator hypothesis. *Neural Computation* 22 (11), 2809–2857.
- Fernando, C., Karishma, K.K., Szathmáry, E., 2008. Copying and evolution of neuronal topology. *PLoS ONE* 3 (11), e3775.
- Fernando, C., Szathmáry, E., 2009a. Chemical, neuronal and linguistic replicators. In: Pigliucci, M., Müller, G. (Eds.), *Towards an Extended Evolutionary Synthesis*. MIT Press, Cambridge, Ma.
- Fernando, C., Szathmáry, E., 2009b. Natural selection in the brain. In: Glatzeder, B., Goel, V., von Müller, A. (Eds.), *Toward a Theory of Thinking*. Springer, Berlin.
- Fodor, J.A., Pylyshyn, Z.W., 1988. Connectionism and cognitive architecture: a critical analysis. *Cognition* 28, 3–71.
- Gutig, F., Sompolinsky, H., 2006. The tempotron: a neuron that learns spike timing-based decisions. *Nature Neuroscience* 9 (3), 420–428.
- Harnad, S., 1990. The symbol grounding problem. *Physica D* 42, 335–346.
- Holland, J.H., Reitman, J.S., 1977. Cognitive systems based on adaptive algorithms. *ACM SIGART Bulletin* 63, 43–49.
- Ikegaya, Y., et al., 2004. Synfire chains and cortical songs: temporal modules of cortical activity. *Science* 304, 559–564.
- Izhikevich, E.M., 2003. Simple model of spiking neurons. *IEEE Transactions on Neural Networks* 14, 1539–1572.
- Izhikevich, E.M., 2006. Polychronization: computation with spikes. *Neural Computation* 18 (2), 245–282.
- Izhikevich, E.M., 2007. Solving the distal reward problem through linkage of stdp and dopamine signaling. *Cerebral Cortex* 17, 2443–2452.
- Izhikevich, E.M., Desai, N.S., 2003. Relating STDP to BCM. *Neural Computation* 15, 1511–1523.
- Izhikevich, E.M., Gally, J.A., Edelman, G.M., 2004. Spike-timing dynamics of neuronal groups. *Cerebral Cortex* 14 (8), 933–944.
- Izhikevich, E.M., Hoppensteadt, F.C., 2009. Polychronous wavefront computations. *International Journal of Bifurcation and Chaos* 19, 1733–1739.
- Karmiloff-Smith, A., 1996. *Beyond modularity: a developmental perspective on cognitive science*. MIT Press, Cambridge, MA.
- Konen, W., Malsburg, C. v. d., 1993. Learning to generalize from single examples in the dynamic link architecture. *Neural Computation* 5 (5), 719–735.
- Love, B.C., 1999. Utilizing time: asynchronous binding. *Advances in Neural Information Processing Systems* 11, 38–44.
- Malsburg, C.v.d., 1999. The what and why of binding: the modeler's perspective. *Neuron* 24, 95–104.
- Marcus, G.F., 2001. *The algebraic mind: integrating connectionism and cognitive science*. MIT Press.
- Marcus, G.F., Vijayan, S., Bandi Rao, S., Vishton, P.M., 1999. Rule learning by seven-month-old infants. *Science* 283 (5398), 77–80.
- Pollack, J.B., 1990. Recursive distributed representations. *Artificial Intelligence* 46 (1), 77–105.
- Ponulak, F., Kasinski, A., 2010. Supervised learning in spiking neural networks with resume: sequence learning, classification and spike-shifting. *Neural Computation* 22 (2), 467–510.
- Seidenberg, M.S., Elman, J.L., 1999. Networks are not 'hidden rules'. *Trends in Cognitive Sciences* 3, 288–289.
- Shastri, L., Ajanagadde, V., 1993. From simple associations to systematic reasoning: a connectionist representation of rules, variables and dynamic bindings. *Behavioural and Brain Sciences* 16, 417–494.
- Song, S., Miller, K.D., et al., 2000. Competitive hebbian learning through spike-timing dependent synaptic plasticity. *Nature Neurosci.* 3, 919–926.
- Steels, L., 2007. The symbol grounding problem is solved, so what's next?. In: De Vega, M.a.G.G.a.G.G. (Ed.), *Symbols, embodiment and meaning*. Academic Press, New Haven.
- Steels, L., Szathmáry, E. 2008. Replicator dynamics and language processing. In: *Proceedings of the 7th International Conference on Evolution of Language (EVOLANG7)*.
- Szathmáry, E., Fernando, C., 2009. Concluding remarks. MIT Press, Cambridge, Ma.
- Unold, O. 2008. Grammar-based classifier system: a universal tool for grammatical inference. In: *Proceedings of the WSEAS Transactions on Computers*.
- Wiskott, L., Malsburg, C.v.d. 1995. Recognizing faces by dynamic link matching. In: *Proceedings of ICANN*.