

A Memory-Based STDP Rule for Stable Attractor Dynamics in Boolean Recurrent Neural Networks

J er mie Cabessa

Laboratory of Mathematical Economics
Universit  Paris 2 – Panth on-Assas
75006 Paris, France

& Institute of Computer Science, Czech Academy of Sciences
E-mail: jeremie.cabessa@u-paris2.fr

Alessandro Villa

NeuroHeuristic Research Group (HEC – ISI)
University of Lausanne
CH-1015 Lausanne, Switzerland
E-mail: alessandro.villa@unil.ch

Abstract—We consider a simplified Boolean model of the basal ganglia-thalamocortical network, and study the effect of a spike-timing-dependent plasticity (STDP) rule on the stabilization of its attractor dynamics. More precisely, we introduce an adaptive STDP rule which constantly updates its learning rate based on the attractors that the network encounters during a window of past time steps. This so-called network memory is assumed to be dynamic: its duration is step-wise increased every time a trigger input pattern is detected, and is decreased otherwise. In this context, we show that well-adjusted trigger inputs can fine tune the network memory and its associated STDP rule in such a way to drive the network into stable and rich attractor dynamics. We discuss how this feature might be related to reward learning processes in the neurobiological context.

I. INTRODUCTION

Boolean recurrent neural networks are simplified neural models composed of interconnections of threshold units (McCulloch and Pitts cells) [1]. At each time step, the state of such a network is a Boolean vector formed by the activation values (quiet or firing) of its composing cells. It has early been established that Boolean neural networks are computationally equivalent to finite state automata [1]–[3]. From a computer science perspective, this result finds its relevance in the possibility to implement finite state machines on parallel hardware [4]. In biology, this feature has been used to study the dynamics of simplified neural networks. In particular, the Boolean approach allows for a complete and systematic analysis of the attractor dynamics of the networks [5].

In neural networks, attractor dynamics or quasi-attractor dynamics have been associated to memories, motor behaviors, perceptions and thoughts [6]–[11]. The chaotic itinerancy between those would then correspond to sequences in thinking, speaking and writing [12]–[16]. In addition, spatiotemporal patterns of discharges—ordered precise spiking patterns that recur above chance level—are likely to be the witnesses of underlying attractor dynamics [17]–[19], and have been observed in relation with specific stimuli or behaviors [20]–[25]. Besides, the different forms of synaptic plasticity are assumed to constitute “the basis for most models of learning, memory and development in neural circuits” [26]. Amongst those mechanisms, spike-timing-dependent plasticity (STDP) refers to the biological Hebbian-like learning processes that

modify the synaptic strengths based on the relative timings of the pre- and post-synaptic spikes.

Based on these considerations, we considered a Boolean model of the basal ganglia-thalamocortical network, and studied the effect of various control parameters on its attractor dynamics [5], [27]–[29]. We showed that feedback regulation, threshold excitability of the cells as well as global and local weight modifications are all features that can, to some extent, combine and compensate with each other in order to improve and stabilize the attractor dynamics of the network. Moreover, we introduced an adaptive STDP rule which, at each time step, updates its learning rate based on the network memory—the set of attractors that the net encounters during a window of past time steps. We showed that, by means of this adaptive STDP rule, the network is capable of improving its attractor dynamics during its computational process [30], [31]

In this paper, we extend these results and study the effect of a more refined STDP rule on the attractor dynamics of the Boolean basal ganglia-thalamocortical network. More precisely, we introduce the concept of a dynamical network memory which is step-wise increased every time the network encounters a specific trigger input pattern, and fades away during the successive time steps as long as no other trigger pattern occurs. We then propose an adaptive STDP rule based on this dynamical memory. In this context, we show that well-adjusted trigger inputs can fine tune the network memory and its associated adaptive STDP rule in such a way to drive the network into stable and rich attractor dynamics. In fact, the frequency of the input patterns may combine with the duration of the memory and lead to efficient attractor dynamics. We discuss how this feature might be related to reward learning processes in the neurobiological context.

II. BOOLEAN RECURRENT NEURAL NETWORKS AND FINITE STATE AUTOMATA

A Boolean recurrent neural network (BRNN) consists of a network of binary neurons, i.e., cells whose activation values are either firing (1) or quiet (0) [1]. The network \mathcal{N} is composed of M input neurons $(u_j)_{j=1}^M$ that receive external signals from the environment, and N internal neurons $(x_j)_{j=1}^N$ connected together recurrently. Given the activation values

of the input neurons $(u_j(t))_{j=1}^M$ and the internal neurons $(x_j(t))_{j=1}^N$ at time t , the activation values of the internal neurons $(x_i(t+1))_{i=1}^N$ at time $t+1$ are given by the following equations:

$$x_i(t+1) = \theta \left(\sum_{j=1}^N a_{ij}(t) \cdot x_j(t) + \sum_{j=1}^M b_{ij}(t) \cdot u_j(t) + c_i(t) \right) \quad (1)$$

for $i = 1, \dots, N$

where $a_{ij}(t)$ is the *synaptic weight* from x_j to x_i , $b_{ij}(t)$ is the *synaptic weight* from u_j to x_i , and $c_i(t)$ is the background activity or *bias* received by x_i , all of them at time t . In addition, θ is the hard-threshold activation function determining the activation value (or neuronal state) of the cells, and defined by

$$\theta(x) = \begin{cases} 0 & \text{if } x < 1 \\ 1 & \text{if } x \geq 1. \end{cases}$$

If the weights and bias a_{ij} , b_{ij} and c_i are not time-dependent, the network is said to be *static*. Otherwise, the network is called *evolving*. A static recurrent neural network is illustrated in Figure 1A.

According to Equation (1), the dynamics of the whole network \mathcal{N} is described by the following system

$$\mathbf{x}(t+1) = f_\theta(\mathbf{A}(t) \cdot \mathbf{x}(t) + \mathbf{B}(t) \cdot \mathbf{u}(t) + \mathbf{c}(t)) \quad (2)$$

where $\mathbf{A}(t) = (a_{ij}(t))$, $\mathbf{B}(t) = (b_{ij}(t))$ and $\mathbf{c}(t) = (c_i(t))$ are the weight matrices and bias vector at time t , respectively, and f_θ denotes the hard-threshold function θ applied componentwise. The tuple of weight and bias matrices $\mathbf{W}(t) = (\mathbf{A}(t), \mathbf{B}(t), \mathbf{c}(t))$ is the (*weight configuration*) of \mathcal{N} at time t . The Boolean vectors

$$\begin{aligned} \mathbf{u}(t) &= (u_1(t), \dots, u_M(t)) \in \mathbb{B}^M \\ \mathbf{x}(t) &= (x_1(t), \dots, x_N(t)) \in \mathbb{B}^N \end{aligned}$$

are the input and internal *states* of \mathcal{N} at time t , respectively. The *dynamics* of \mathcal{N} over u , denoted by $\mathcal{N}(u)$, is the sequence of successive internal states encountered by \mathcal{N} while processing some input stream $u = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2)\dots$, i.e. $\mathcal{N}(u) = \mathbf{x}(0)\mathbf{x}(1)\mathbf{x}(2)\dots$. Any finite sequence of the form $\mathbf{x}(0)\mathbf{x}(1)\dots\mathbf{x}(k-1) \in (\mathbb{B}^N)^k$ can naturally and unambiguously be extended into an infinite one by adding infinitely many null states $\mathbf{0}$ after $\mathbf{x}(k-1)$. Hence, without loss of generality, we can assume that any dynamics of \mathcal{N} is infinite. Finally, the *evolution* of \mathcal{N} over u , denoted $\mathcal{E}(u)$, is the sequence of successive configurations encountered by \mathcal{N} while processing input u , i.e. $\mathcal{E}(u) = \mathbf{W}(0)\mathbf{W}(1)\mathbf{W}(2)\dots$.

An *attractor* of \mathcal{N} is a set of internal states into which the dynamics of the network can get trapped, but not necessarily in a periodic manner. Formally, a set $X = \{\mathbf{x}_0, \dots, \mathbf{x}_k\} \subseteq \mathbb{B}^N$ is an attractor of \mathcal{N} if there exists some finite or infinite input stream $u = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2)\dots$ and some index $i_0 \in \mathbb{N}$ such that the corresponding dynamics $\mathcal{N}(u) = \mathbf{x}(0)\mathbf{x}(1)\mathbf{x}(2)\dots$ satisfies $\mathbf{x}(i) \in X$, for all $i \geq i_0$.

In the case of static Boolean networks \mathcal{N} , the weight matrices and biases do not change over time, which is formally

expressed as $\mathbf{W}(t) = \mathbf{W}(t')$ for all $t, t' > 0$. In this context, it is established that static Boolean recurrent neural networks (BRNN) are computationally equivalent to finite state automata (FSA) [1]–[3]. More precisely, for any network \mathcal{N} , there exists a corresponding automaton \mathcal{A} that can simulate it (in a precise sense), and vice versa. In particular, given some network \mathcal{N} , the construction of an equivalent automaton \mathcal{A} is given as follows: the nodes of \mathcal{A} correspond to the internal states of \mathcal{N} , and there exists an edge from node \mathbf{x} to \mathbf{x}' labelled by \mathbf{u} in \mathcal{A} if and only if \mathcal{N} moves from state \mathbf{x} to \mathbf{x}' when receiving the Boolean input \mathbf{u} . This construction is illustrated in Figure 1. According to this construction, the possible *dynamics* of \mathcal{N} correspond precisely to the various *paths* in the graph of \mathcal{A} . As a consequence, the cyclic dynamics—i.e., the *attractors*—of \mathcal{N} correspond precisely to the cyclic paths—i.e., the *cycles*—of \mathcal{A} [5]. Therefore, the set of attractors of any static Boolean network can be computed explicitly: it suffices to construct the corresponding finite automaton, and then list all of its cycles. The set and number of attractors in a static network \mathcal{N} are denoted by \bar{A} and n , respectively, where $n = |\bar{A}|$.

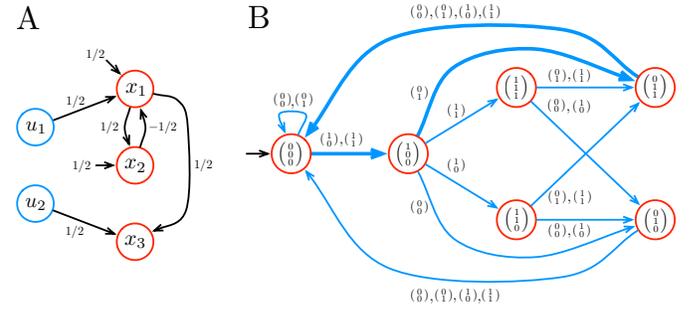


Fig. 1: **A.** Boolean neural network \mathcal{N} with 2 input cells (blue) and 3 internal cells (red). **B.** The finite automaton \mathcal{A} associated to the network \mathcal{N} . The nodes of \mathcal{A} are the internal states of \mathcal{N} , and there is an edge from node \mathbf{x} to node \mathbf{x}' labelled by \mathbf{u} if and only if network \mathcal{N} switches from state \mathbf{x} to state \mathbf{x}' when receiving input \mathbf{u} . The cycles in the graph of \mathcal{A} correspond to the attractors of \mathcal{N} . For instance, the boldface cycle corresponds to the attractor $X = \{(0, 0, 0)^T, (1, 0, 0)^T, (0, 1, 1)^T\}$.

The case of evolving Boolean networks is slightly more complex. In this context, any configuration $\mathbf{W}(t) = (\mathbf{A}(t), \mathbf{B}(t), \mathbf{c}(t))$ of \mathcal{N} corresponds to a specific static network, and thus, can be converted into a corresponding finite automaton $\mathcal{A}(t)$ in the way described previously. From this automaton, one can compute the *set of attractors of \mathcal{N} at time t* (in the way described previously also), denoted by \bar{A}_t , and the *number of attractors of \mathcal{N} at time t* , given by $n_t = |\bar{A}_t|$. In other words, the attractors of \mathcal{N} at time t are the attractors that the network could potentially visit while staying in the static weight configuration $\mathbf{W}(t)$. If the configuration $\mathbf{W}(t)$ changes over time, then so does the associated automaton $\mathcal{A}(t)$, and thus also possibly the set and number of attractors \bar{A}_t and n_t . According to these considerations, for any input stream $u = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2)\dots$, the evolution $\mathcal{E}(u) = \mathbf{W}(0)\mathbf{W}(1)\mathbf{W}(2)\dots$ of \mathcal{N} induces a

corresponding sequence of finite automata $\mathcal{A}(0)\mathcal{A}(1)\mathcal{A}(2)\dots$, which in turn gives rise to two sequences of sets and numbers of attractors $(\bar{A}_t)_{t \geq 0}$ and $(n_t)_{t \geq 0}$, respectively. Note that the set and number of attractors change across time as the network processes input stream u . The varying attractor dynamics of an evolving Boolean neural network processing a given input stream is illustrated in Figure 2.

In this work, we assume that some relevant aspects of the computational complexity of neural networks are related to their attractor dynamics, and more specifically, to the number of attractor that they possess. Consequently, in the sequel, we will focus on the sequence (i.e., the variation) of numbers of attractors $(n_t)_{t \geq 0}$ that the network encounters as it processes its input stream.

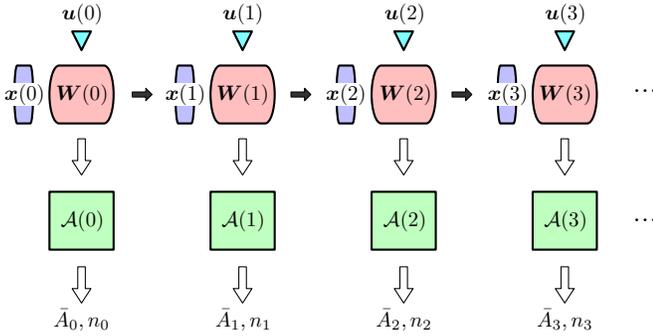


Fig. 2: An evolving Boolean recurrent neural network \mathcal{N} processing an input stream $u = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2)\dots$. The blue triangles represent the successive elements of u . The purple elements are the successive states forming the dynamics $\mathcal{N}(u) = \mathbf{x}(0)\mathbf{x}(1)\mathbf{x}(2)\dots$. The red elements are the successive (weight) configurations of the evolution $\mathcal{E}(u) = \mathbf{W}(0)\mathbf{W}(1)\mathbf{W}(2)\dots$. The conjunction of each input $\mathbf{u}(t)$, state $\mathbf{x}(t)$ and configuration $\mathbf{W}(t)$ determines the next state $\mathbf{x}(t+1)$ (cf. Equation (2)). The green elements form the sequence of finite automata $\mathcal{A}(0)\mathcal{A}(1)\mathcal{A}(2)\dots$ associated to the successive configurations of \mathcal{N} . To each automaton $\mathcal{A}(t)$ corresponds a set and number of attractors \bar{A}_t and n_t , respectively. As the network processes input stream u , its set and number of attractors varies across time.

III. DYNAMIC MEMORY AND ADAPTIVE STDP RULE

In our context, the *memory* of a network consists of an integer $m \geq 0$ representing a time interval. If the network has a memory of m , then at any time steps t , it can “remember” the attractors that were encountered during the m past time steps of its dynamics. In this work, we introduce the concept of a *dynamic memory*, i.e., a memory that can vary over time. Accordingly, the network might have a more or less extended knowledge of its past attractors as it progresses along its dynamics (cf. Figure 3).

Formally, the *dynamic memory* of an evolving Boolean neural network \mathcal{N} is a sequence of positive integers $(m_t)_{t \geq 0}$, where each m_t represents the memory at time step t . For any input stream $u = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2)\dots$ with corresponding numbers of attractors $(n_t)_{t \geq 0}$ and for any dynamic memory

$(m_t)_{t \geq 0}$, the *dynamic memory content* of \mathcal{N} is the sequence of stacks¹ $(M_t)_{t \geq 0}$ recursively defined as follows:

$$\begin{aligned} M_0 &= \begin{cases} n_0 & \text{if } m_0 > 0 \\ \emptyset & \text{if } m_0 = 0 \end{cases} \\ M_{t+1} &= [M_t \widehat{\ } n_{t+1}]_{m_{t+1}} \end{aligned} \quad (3)$$

where $s \widehat{\ } x$ denotes the sequence obtained by concatenating the elements of s with element x , and $[s]_n$ is the sequence of the last n elements of s if $|s| > n$ and $[s]_n = s$ otherwise. In other words, the memory content M_{t+1} is obtained by taking the last m_{t+1} elements of the content M_t concatenated with n_{t+1} . Hence, each memory content M_t is a stack of length less or equal to m_t . In the sequel, the minimum and maximum elements of any memory content M_t are denoted by $\min(M_t)$ and $\max(M_t)$, respectively. The dynamic memory of a network is illustrated in Figure 3.

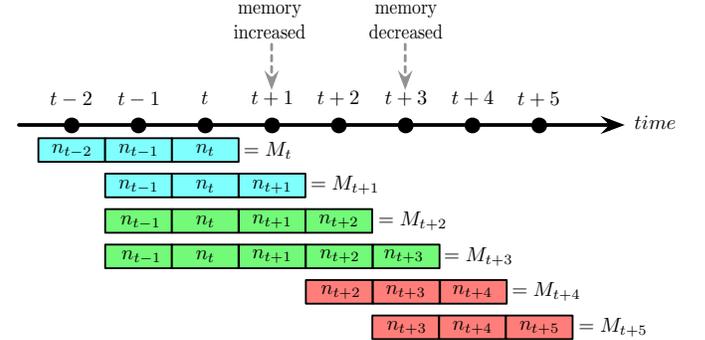


Fig. 3: Evolution of the dynamic memory $(m_t)_{t \geq 0}$ of a neural network along its dynamics. The successive dynamic memory contents $(M_t)_{t \geq 0}$ are represented by the colored striped. Each M_t is a stack of the form \emptyset or $n_{t-k}\dots n_t$, for some $k \geq 0$. At time steps t and $t+1$, the dynamic memory satisfies $m_t = m_{t+1} = 3$ (blue pattern). This means that the network remembers the number of attractors of the last 3 time steps (including the current time step). At next time steps $t+2$ and $t+3$, one has $m_{t+2}, m_{t+3} \geq 5$ (green pattern). In this case, the network remembers more and more number of attractors, up to the last 5 time steps. At time steps $t+4$ and $t+5$, one has $m_{t+4} = m_{t+5} = 3$ (red pattern). In these cases, the network remembers the number of attractors of the last 3 time steps

A *spike-timing dependent plasticity (STDP) rule* modifies the synaptic weights $a_{ij}(t)$ according to the spiking patterns of the presynaptic and postsynaptic cells x_j and x_i . Usually, the synaptic weight $a_{ij}(t)$ is increased (resp. decreased) by a certain amount λ , called the *learning rate*, if the presynaptic cell x_j spikes before (resp. after) the postsynaptic cell x_i [32]. Recently, an *adaptive STDP rule*—where the learning rate varies instead of remaining fixed over time—has been proposed [31], [33]. Here, this adaptive STDP rule is generalized to the context of dynamic memories.

¹A stack is simply a finite sequence $s = s_0s_1\dots s_n$. The empty stack is denoted by \emptyset , and a single element stack is denoted as s_0 .

We consider the following *adaptive STDP* rule. For any weight a_{ij} , let $I_{ij} = [b_{ij}, b'_{ij}] \subset \mathbb{R}$ be a real interval. The variation of a_{ij} is then given as follows:

$$a_{ij}(t+1) = \begin{cases} b_{ij} & \text{if } R < b_{ij} \\ R & \text{if } b_{ij} \leq R \leq b'_{ij}, \\ b'_{ij} & \text{if } R > b'_{ij} \end{cases}, \quad \text{where}$$

$$R = a_{ij}(t) + \lambda(t) [x_i(t+1) \cdot x_j(t) - x_i(t) \cdot x_j(t+1)] \quad (4)$$

Note that the *learning rate* $\lambda(t)$ is time-dependent. At time $t+1$, the synaptic weight $a_{ij}(t)$ is increased (resp. decreased) by $\lambda(t)$ (up to the bounds of $I_{ij} = [b_{ij}, b'_{ij}]$) if and only if the presynaptic cell x_j has spiked one time step before (resp. after) the postsynaptic cell x_i .

In our context, the evolution of the adaptive learning rate $\lambda(t)$ is based on the memory of the network—which is itself dynamic. More precisely, $\lambda(t)$ is defined as the image of n_t by the linear interpolation between two points $(\min(M_t), \lambda_+)$ and $(\max(M_t), \lambda_-)$, where $\lambda_-, \lambda_+ \in \mathbb{R}$ are two bounds such that $\lambda_- < \lambda_+$. Formally,

$$\lambda(t) = \begin{cases} \lambda_+ + \frac{(n_t - \min(M_t))(\lambda_- - \lambda_+)}{\max(M_t) - \min(M_t)} & \text{if } \min(M_t) < n_t < \max(M_t) \\ \lambda_+ & \text{otherwise.} \end{cases} \quad (5)$$

The computation of $\lambda(t)$ is illustrated in Figure 4. The learning rate $\lambda(t)$ has to be understood as follows. If $n_t = \min(M_t)$ (resp. $n_t = \max(M_t)$), then it means that the current number of attractors of the network is at a minimal (resp. maximal) level, i.e., n_t corresponds to the minimal (resp. maximal) numbers of attractors remembered by the network during the m_t last time steps. In this case, $\lambda(t) = \lambda_+$ (resp. $\lambda(t) = \lambda_-$). This large (resp. low) learning rate will induce large (resp. low) variations of the synaptic weights (cf. Equation (4)) with the aim of destabilizing (resp. stabilizing) the current weight configuration of the network. If $\min(M_t) < n_t < \max(M_t)$, then $\lambda_+ > \lambda(t) > \lambda_-$ according to the linear interpolation. The closer n_t is to $\min(M_t)$ (resp. to $\max(M_t)$), the closer $\lambda(t)$ is to λ_+ (resp. to λ_-). If $\min(M_t) = \max(M_t)$, the network has settled into the same attractor dynamics during the m_t last steps. In this case, we set $\lambda(t) = \lambda_+$ with the aim of destabilizing the current configuration.

Observe that, at every time step t , if the network has no memory, i.e., $m_t = 0$, then $M_t = \emptyset$ (cf. Equation (3)), thus $\lambda(t) = \lambda_+$ for all $t > 0$ (cf. Equation (5)), and therefore, the network is subjected to a (classical) *fixed-rate* STDP rule. By contrast, whenever the memory is strictly positive (i.e. $m_t > 0$), the network is subjected to an *adaptive* STDP rule whose learning rate depends on the current memory length m_t and number of attractors n_t . This *adaptive* feature is crucial to the improvement and stabilization of the attractor dynamics of the network [30], [31].

In the sequel, the evolution of the network memory will be based on some trigger patterns that the network encounters throughout its dynamics. In fact, the memory will be significantly increased every time the network encounters a specific input pattern, and fades away during the successive time steps as long as not other trigger pattern occurs.

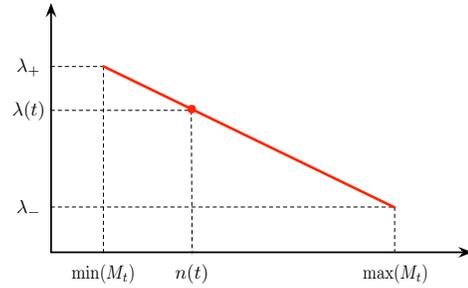


Fig. 4: The adaptive learning rate $\lambda(t)$ is the image of n_t by the linear interpolation between the two points $(\min(M_t), \lambda_+)$ and $(\max(M_t), \lambda_-)$. As time progresses, the values $\min(M_t)$ and $\max(M_t)$ evolve along the x -axis. The interpolation lines (red) varies in consequence.

IV. BOOLEAN MODEL OF THE BASAL GANGLIA-THALAMOCORTICAL NETWORK

Here, we consider a simplified Boolean model of the Basal Ganglia-Thalamocortical (BGT) network, illustrated in Figure 5A. The circuit processes sensorimotor information at various levels of integration in brain activity [34]–[38]. We assume that each brain area is represented by a Boolean node in this circuit. The pattern of connectivity between the nodes is based on the wealth of data reported in the literature, and described in more details elsewhere [5]. The 9 nodes included in the current BGT network correspond to the superior colliculus (node 1), the thalamus (node 2), the thalamic reticular nucleus (node 3), the output nuclei of the basal ganglia formed by the GABAergic projection neurons of the intermediate part of the pallidum and of the substantia nigra pars reticulata (node 4), the subthalamic nucleus (node 5), the external part of the pallidum (node 6), the striatopallidal (Str-D2, node 7) and the striatonigral (Str-D1, node 8) components of the striatum, and the cerebral cortex (node 9). The closed-loop architecture of the network is implemented via feedback connections from the efferent outputs to the input (IN, node 0). The weighting pattern of the BGT network (Figure 5A) is given by the adjacency matrix of Table 1. An analysis of its corresponding finite automaton (Figure 5B) shows that this network possesses 22 attractors.

Table 1: Weight matrix of the Boolean model of the BGT network of Figure 5A.

# Node	Target									
	0	1	2	3	4	5	6	7	8	9
0 IN	.	1	1
1 SC	int ₁	.	1
2 Thalamus	.	.	.	1	.	1	1	1	1	1
3 NRT	.	.	-1
4 GPi/SNr	.	-1	-1	-1
5 STN	2	.	2	.	.	2
6 GPe	.	.	.	-1/2	-1/2	-1/2	.	-1/2	-1/2	.
7 Str-D2	-1	.	.	.
8 Str-D1	-1/2	.	-1/2	.	.	.
9 C. Cortex	int ₂	1/2	1	1/2	.	1/2	.	1/2	1/2	.

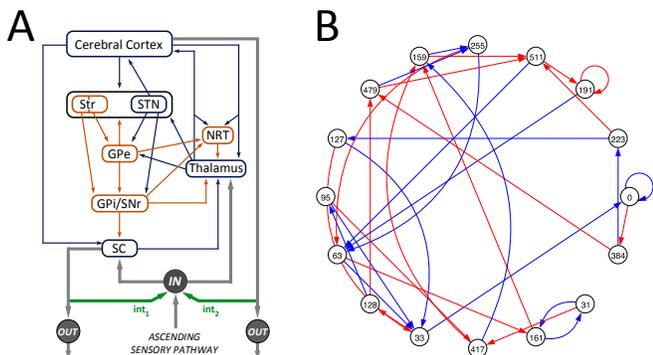


Fig. 5: **A.** Simplified Boolean model of the BGT network. Each brain area is represented by a single Boolean unit. The network is formed by 10 Boolean nodes: IN, SC, Thalamus, NRT, Cerebral Cortex, Str-D1, Str-D2, STN, GPe, GPi/SNr. The excitatory and inhibitory pathways are labeled in blue and orange, respectively. **B.** Finite automaton associated to the Boolean model of the BGT network of panel A with weight matrix given in Table 1. Each node of the automaton is a Boolean state of the network. There is a blue or red transition from node i to node j if and only if the network switches from state i to state j when receiving input 0 or 1, respectively. Modified figure from [31].

V. RESULTS

By means of computer simulations, we study the effect of our adaptive STDP rule on the stabilization of the attractor dynamics of the BGT network.

At the beginning of each simulation, every non-zero weight a_{ij} of Table 1 was jittered by a random uniform noise $\epsilon_{ij} \sim \mathcal{U}(-0.1, 0.5)$ in order to introduce some variability in the original weights. In addition, based on empirical considerations, the weight intervals were chosen as $I_{ij} = [(a_{ij} + \epsilon_{ij}) - 0.2; (a_{ij} + \epsilon_{ij}) + 0.6]$, for all $a_{ij} > 0$. Then, we provided the BGT network with a random input stream $u = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2) \dots$ of length 3000 interspersed with occurrences of a specific *trigger pattern* of size 10. The core idea of this study resides in the assumption that the dynamic memory of the network is reinforced by the occurrences of the trigger pattern. Accordingly, we assume that the network increases its memory by 150 every time a trigger pattern is encountered. Afterwards, at each successive time step, the memory is decreased by 1 as long as no other trigger pattern is met. Formally, the dynamic memory $(m_t)_{t \geq 0}$ is recursively defined as follows:

$$m_0 = 0$$

$$m_{t+1} = \begin{cases} m_t + 150 & \text{if } \mathbf{u}(t-8) \dots \mathbf{u}(t+1) \text{ is} \\ & \text{the trigger pattern} \\ \max(m_t - 1, 0) & \text{otherwise} \end{cases} \quad (6)$$

Furthermore, the extreme values of adaptive plasticity rates are set to $\lambda_- = 0.0075$ and $\lambda_+ = 0.15$.

At every time step, we computed the (largest strongly connected component of) the finite automaton associated to the BGT network, and in turn, the current number of attractors of

this latter, as described in Section II and illustrated in Figure 2. We performed several simulations using the same random seed, namely the same random input pattern u and jittered weights $a_{ij} + \epsilon_{ij}$, but for different number of occurrences of the trigger pattern. The results are reported in Figure 6.

The four plots of Figure 6 depict the evolutions in the number of attractors of the BGT network processing a same input stream interspersed with 5, 9, 19 and 26 trigger patterns, respectively. It is clearly seen that the increase in the number of the trigger patterns induces an increase in the stabilization—or equivalently, a decrease in the fluctuation—of the network’s number of attractors. This feature is rendered explicit in Table 2, which shows that the average time during which the attractor dynamics of the network remains stable clearly increases as the number of trigger patterns is incremented. In fact, each time a trigger pattern is received, the corresponding increase in memory (cf. Equation (6)) enables the network to “constitute a past” of its attractor dynamics, and in turn, to adjust its learning rate so as to stabilize itself in the maximal number of attractors that it has remembered so far (cf. Equation (5)).

Amongst the four simulations, the most profitable is the third where many stable periods of 41 attractors happen, as well as 4 periods of 209 attractors. But the stability of the latter higher regimes cannot be maintained during most part of the dynamics. By contrast, in the fourth simulation, the attractor dynamics is much more stable, but the higher regimes cannot be reached anymore. In this case, the overabundance of trigger patterns seems to favor an extreme stability at the expense of some efficient variability.

The third simulation of Figure 6 is illustrated in more details in Figure 7. The fluctuation of the number of attractors (black trace), memory (red trace) and learning rate (green trace) are represented. We clearly see how the memory accumulates as the trigger patterns occur and fades away between those events. We also remark the correlation between the low (resp. high) values of $\lambda(t)$ and the stable (resp. unstable) attractor regimes. This shows that the memory-based fluctuations of the learning rate manage to stabilize or destabilize the attractor regimes.

Table 2: Statistics about the fluctuation in the number of attractors for the four simulations displayed in Figure 6. The number of fluctuations is the number of times that the number of attractors changes during the network dynamics. The maximum and average times refer to the maximal and average continuous periods during which the network keeps a same number of attractors.

Figure 6	plot 1	plot 2	plot 3	plot 4
# trigger patterns	5	9	19	26
# fluctuations	707	584	187	71
maximum time	212	204	255	963
average time	4.24	5.13	16	34.34
Same statistics after the first trigger pattern has occurred				
# fluctuations	357	525	123	12
maximum time	212	204	255	963
average time	5.26	5.31	22.62	185.67

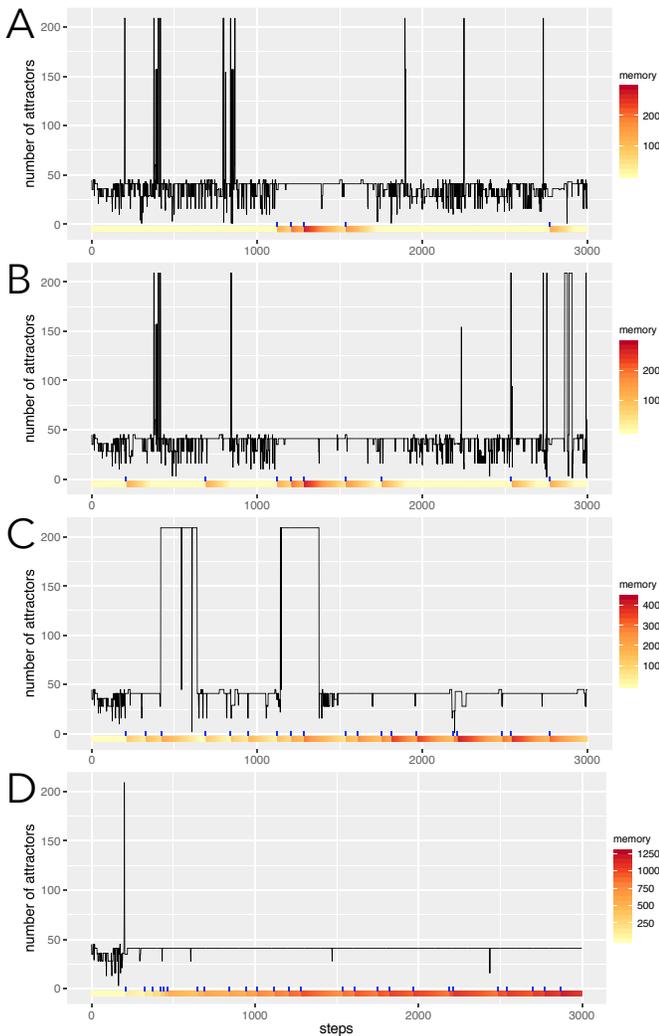


Fig. 6: Evolution of the number of attractors of the BGT network throughout its dynamic. The BGT network is subjected to a random input stream of length 3000. During this process, a trigger pattern of length 10 is inserted x times at random time steps, where x equals 5, 9, 19 and 26 in panel *A*, *B*, *C* and *D*, respectively. The occurrences of the trigger pattern are depicted by the blue segments. Every time a trigger pattern occurs, the memory of the network is increased by 150, and then decreased by 1 at each successive time step as long as no other trigger pattern is encountered. The value of the network memory is represented as a color bar at the bottom of each plot (whose scales are different). Notice that the more frequent the trigger pattern occurs, the more stable the attractor dynamics.

VI. CONCLUSION

The rationale underlying this study is that the number of attractors would be significantly related to some aspect of complexity in the information processing achieved by a Boolean recurrent neural network [31], [33]. In fact, attractor dynamics in the brain are likely to be associated with oscillatory patterns observed throughout the basal ganglia-thalamocortical circuit. Information about duration and identity of a neural representation can be extracted from the relative phase of

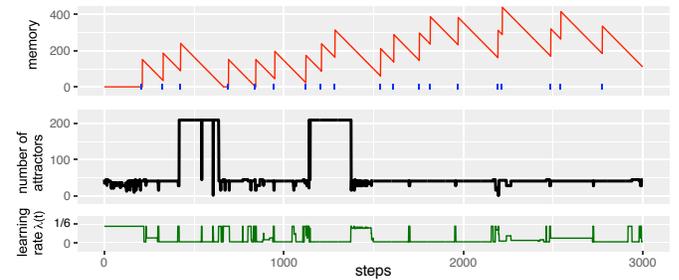


Fig. 7: Details of the third simulation of Figure 6. The black trace represents the evolution of the number of attractors over time (as in Figure 6). The red trace represents the variation of the dynamic memory over time. After every occurrence of a trigger pattern (blue dot), the memory is increased by 150. The green trace represents the fluctuation of the learning rate $\lambda(t)$ of the STDP rule. Note the correlation between the low (resp. high) values of $\lambda(t)$ and the stable (resp. unstable) regions of the attractor dynamics: when $\lambda(t)$ is low, the STDP rule has almost no influence on the weights, and hence the number of attractors of the network remains constant.

the oscillations [39]. Synaptic plasticity mechanisms can lead to changes in synchronized activity, and in turn, affect the dynamics of learning [40].

The current work introduces the concept of *dynamic memory* and assumes the existence of an *adaptive STDP rule* whose variation depends on this memory. In short, the network constantly updates the memory it has about its attractor dynamics, and its synaptic plasticity mechanism changes accordingly. This general approach underlying this idea is that the conjunction of *unsupervised* and *self-organizing* processes should lie at the core of brain computations, and are necessary for the achievement of efficient representations of information. Here, we assume the existence of a pattern recognition system that would trigger a reward-driven process, which in turn would induce changes in the network memory. Afterwards, based on its current memory, the network modifies its learning rate by comparing the richness of its current and remembered attractor dynamics. This possibility for the network to assess its attractor dynamics would be achieved by means of biochemical signals, rather than by the possibility to really count the number of attractors.

Cortico-basal ganglia-thalamo-cortical loops were recognized to play a crucial role in temporal sequence storage and regeneration [41]. The biological mechanism underlying these phenomena are the relations between memory storage, synaptic plasticity and calcium dynamics [42], as well as the perturbation of working memory timing mediated by dopaminergic reward pathways [43]. This dynamic memory model is supported by evidence of modifiable sensory responses modulated by reinforcement signals occurring in the basal ganglia [44]. Reward information via dopaminergic and other modulatory pathways (e.g., cholinergic, serotonergic) of the basal ganglia is integrated with the control of voluntary movements performed by a cooperative activation of the striatopallidal and striatonigral pathways [45], [46]. Along the

same line, it was observed that dysfunctions of the reward signals are distinctive feature of the addicted brain [47].

In conclusion, the dynamic memory-based STDP rule introduced here provides a model for interpreting features of learning dynamics observed in experimental studies. In this work, we have assumed the existence of pattern recognition and reward circuits. In future work, we intend to elaborate a more complete model with those circuits taken into consideration.

ACKNOWLEDGMENT

Supports from DARPA – Lifelong Learning Machines (L2M) program, cooperative agreement No. HR0011-18-2-0023, as well as from the Czech Science Foundation, grant No. 19-05704S are gratefully acknowledged.

REFERENCES

- [1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.
- [2] S. C. Kleene, "Representation of events in nerve nets and finite automata," in *Automata Studies*, C. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton University Press, 1956, pp. 3–41.
- [3] M. L. Minsky, *Computation: finite and infinite machines*. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1967.
- [4] B. G. Horne and D. R. Hush, "Bounds on the complexity of recurrent neural network implementations of finite state machines," *Neural Networks*, vol. 9, no. 2, pp. 243–252, 1996.
- [5] J. Cabessa and A. E. P. Villa, "An attractor-based complexity measurement for boolean recurrent neural networks," *PLoS ONE*, vol. 9, no. 4, pp. e94204+, 2014.
- [6] W. A. Little, "The existence of persistent states in the brain," *Mathematical biosciences*, vol. 19, pp. 101–120, 1974.
- [7] W. A. Little and G. L. Shaw, "Analytical study of the memory storage capacity of a neural network," *Mathematical biosciences*, vol. 39, pp. 281–290, 1978.
- [8] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, pp. 2554–2558, 1982.
- [9] —, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences*, vol. 81, no. 10, pp. 3088–3092, 1984.
- [10] D. J. Amit, *Modeling brain function: The world of attractor neural networks*. Cambridge University Press, 1992.
- [11] C. Eliasmith, "A unified approach to building and controlling spiking attractor networks," *Neural Comput.*, vol. 17, no. 6, pp. 1276–1314, 2005.
- [12] K. Ikeda, "Chaotic itinerancy," *International Symposia on Information Sciences (ISKIT) '92*, 1992, kyoto Univ.
- [13] I. Tsuda, "Chaotic itinerancy as a dynamical basis of hermeneutics of brain and mind," *World Futures*, vol. 32, pp. 167–185, 1991.
- [14] —, "Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems," *Behav. Brain Sci.*, vol. 24, no. 5, pp. 793–847, 2001.
- [15] —, "Hypotheses on the functional roles of chaotic transitory dynamics," *Chaos*, vol. 19, pp. 015113-1 – 015113-10, 2009.
- [16] K. Kaneko and I. Tsuda, "Chaotic itinerancy," *Chaos*, vol. 13, no. 3, pp. 926–936, 2003.
- [17] A. Celletti and A. Villa, "Determination of chaotic attractors in the rat brain," *Journal of Statistical Physics*, vol. 84, no. 5, pp. 1379–1385, 1996.
- [18] A. Celletti and A. E. Villa, "Low-dimensional chaotic attractors in the rat brain," *Biol. Cybern.*, vol. 74, no. 5, pp. 387–393, 1996.
- [19] A. E. P. Villa, I. V. Tetko, A. Celletti, and A. Riehle, "Chaotic dynamics in the primate motor cortex depend on motor preparation in a reaction-time task," *Current Psychology of Cognition*, vol. 17, pp. 763–780, 1998.
- [20] M. Abeles, H. Bergman, E. Margalit, and E. Vaadia, "Spatiotemporal firing patterns in the frontal cortex of behaving monkeys," *J. Neurophysiol.*, vol. 70, no. 4, pp. 1629–1638, 1993.
- [21] Y. Prut, E. Vaadia, H. Bergman, I. Haalman, H. Slovin, and M. Abeles, "Spatiotemporal structure of cortical activity: Properties and behavioral relevance," *Journal of Neurophysiology*, vol. 79, no. 6, pp. 2857–2874, 1998.
- [22] A. E. Villa and M. Abeles, "Evidence for spatiotemporal firing patterns within the auditory thalamus of the cat," *Brain Res*, vol. 509, no. 2, pp. 325–327, 1990.
- [23] A. E. P. Villa, "Empirical Evidence about Temporal Structure in Multi-unit Recordings," in *Time and the brain*, ser. Conceptual Advances in Brain Research, R. Miller, Ed. Amsterdam, The Netherlands: Harwood Academic, 2000, vol. 3, ch. 1, pp. 1–51.
- [24] E. Vaadia, I. Haalman, M. Abeles, H. Bergman, Y. Prut, H. Slovin, and A. Aertsen, "Dynamics of neuronal interactions in monkey cortex in relation to behavioural events," *Nature*, vol. 373, no. 6514, pp. 515–518, 1995.
- [25] T. Shmiel, R. Drori, O. Shmiel, Y. Ben-Shaul, Z. Nadasdy, M. Shemesh, M. Teicher, and M. Abeles, "Neurons of the cerebral cortex exhibit precise interspike timing in correspondence to behavior," *Proc. Natl. Acad. Sci. USA*, vol. 102, no. 51, pp. 18655–18657, 2005.
- [26] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: taming the beast," *Nat. Neurosci.*, vol. 3 Suppl., pp. 1178–1183, 2000.
- [27] J. Cabessa and A. E. P. Villa, "Attractor-based complexity of a boolean model of the basal ganglia-thalamocortical network," in *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*. IEEE, 2016, pp. 4664–4671.
- [28] —, "Attractor dynamics driven by interactivity in boolean recurrent neural networks," in *Artificial Neural Networks and Machine Learning - ICANN 2016 - 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part I*, ser. Lecture Notes in Computer Science, A. E. P. Villa, P. Masulli, and A. J. P. Rivero, Eds., vol. 9886. Springer, 2016, pp. 115–122.
- [29] —, "Interactive control of computational power in a model of the basal ganglia-thalamocortical circuit by a supervised attractor-based learning procedure," in *Artificial Neural Networks and Machine Learning - ICANN 2017 - 26th International Conference on Artificial Neural Networks, Alghero, Italy, September 11-14, 2017, Proceedings, Part I*, ser. Lecture Notes in Computer Science, A. Lintas, S. Rovetta, P. F. M. J. Verschuer, and A. E. P. Villa, Eds., vol. 10613. Springer, 2017, pp. 334–342. [Online]. Available: https://doi.org/10.1007/978-3-319-68600-4_39
- [30] —, "An STDP rule for the improvement and stabilization of the attractor dynamics of the basal ganglia-thalamocortical network," in *Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III*, ser. Lecture Notes in Computer Science, V. Kurková, Y. Manolopoulos, B. Hammer, L. S. Iliadis, and I. Maglogiannis, Eds., vol. 11141. Springer, 2018, pp. 693–702. [Online]. Available: https://doi.org/10.1007/978-3-030-01424-7_68
- [31] —, "Attractor dynamics of a boolean model of a brain circuit controlled by multiple parameters," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 10, p. 106318, 2018. [Online]. Available: <https://doi.org/10.1063/1.5042312>
- [32] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: a Hebbian learning rule," *Annu. Rev. Neurosci.*, vol. 31, pp. 25–46, 2008.
- [33] J. Cabessa and A. E. P. Villa, "An STDP Rule for the Improvement and Stabilization of the Attractor Dynamics of the Basal Ganglia-Thalamocortical Network," *Lect Notes Comput Sci*, vol. 11141, pp. 693–702, 2018.
- [34] G. Alexander and M. Crutcher, "Functional architecture of basal ganglia circuits: neural substrates of parallel processing," *Trends Neurosci.*, vol. 13, no. 7, pp. 266–271, 1990.
- [35] J. E. Hoover and P. L. Strick, "Multiple output channels in the basal ganglia," *Science*, vol. 259, no. 5096, pp. 819–821, Feb 1993.
- [36] J. D. Kropotov and S. C. Etlinger, "Selection of actions in the basal ganglia thalamocortical circuits: review and model," *International Journal of Psychophysiology*, vol. 31, no. 3, pp. 197–217, 1999.
- [37] A. Leblois, T. Boraud, W. Meissner, H. Bergman, and D. Hansel, "Competition between feedback loops underlies normal and pathological dynamics in the basal ganglia," *J. Neurosci.*, vol. 26, no. 13, pp. 3567–3583, 2006.
- [38] A. Lintas, I. G. Silkis, L. Albéri, and A. E. P. Villa, "Dopamine deficiency increases synchronized activity in the rat subthalamic nucleus," *Brain Res*, vol. 1434, pp. 142–151, 2012.

- [39] C. Lustig, M. S. Matell, and W. H. Meck, "Not "just" a coincidence: frontal-striatal interactions in working memory and interval timing," *Memory*, vol. 13, no. 3-4, pp. 441-448, 2005.
- [40] A. Gruart, J. M. Delgado-García, and A. Lintas, "Effect of parvalbumin deficiency on distributed activity and interactions in neural circuits activated by instrumental learning," in *Advances in Cognitive Neurodynamics (V)*, R. Wang and X. Pan, Eds. Singapore: Springer Singapore, 2016, pp. 111-117.
- [41] J. G. Taylor and N. R. Taylor, "Analysis of recurrent cortico-basal ganglia-thalamic loops for working memory," *Biol Cybern*, vol. 82, no. 5, pp. 415-32, May 2000.
- [42] J. Jędrzejewska-Szmek, S. Damodaran, D. B. Dorman, and K. T. Blackwell, "Calcium dynamics predict direction of synaptic plasticity in striatal spiny projection neurons," *Eur J Neurosci*, vol. 45, no. 8, pp. 1044-1056, 04 2017.
- [43] J. T. Coull, H. J. Hwang, M. Leyton, and A. Dagher, "Dopamine precursor depletion impairs timing in healthy volunteers by attenuating activity in putamen and supplementary motor area," *J Neurosci*, vol. 32, no. 47, pp. 16704-15, Nov 2012.
- [44] H. Nakahara, S. Amari Si, and O. Hikosaka, "Self-organization in the basal ganglia with modulation of reinforcement signals," *Neural Comput.*, vol. 14, no. 4, pp. 819-844, 2002.
- [45] Y. Isomura, T. Takekawa, R. Harukuni, T. Handa, H. Aizawa, M. Takada, and T. Fukai, "Reward-modulated motor information in identified striatum neurons," *J Neurosci*, vol. 33, no. 25, pp. 10209-10220, Jun 2013.
- [46] X. Jin, F. Tecuapetla, and R. M. Costa, "Basal ganglia subcircuits distinctively encode the parsing and concatenation of action sequences," *Nat Neurosci*, vol. 17, no. 3, pp. 423-430, Mar 2014.
- [47] S. Spiga, A. Lintas, and M. Diana, "Altered Mesolimbic Dopamine System in THC Dependence," *Curr Neuropharmacol*, vol. 9, no. 1, pp. 200-4, Mar 2011.