

# A neuronal device for the control of multi-step computations

Ariel Zylberberg, Luciano Paz, Pieter R. Roelfsema, Stanislas Dehaene, Mariano Sigman

▶ **To cite this version:**

Ariel Zylberberg, Luciano Paz, Pieter R. Roelfsema, Stanislas Dehaene, Mariano Sigman. A neuronal device for the control of multi-step computations. Papers in physics, 2013, 5, pp.050006. 10.4279/PIP.050006 . cea-02933065

**HAL Id: cea-02933065**
**https://cea.hal.science/cea-02933065**

Submitted on 8 Sep 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Papers in Physics, vol. 5, art. 050006 (2013)

www.papersinphysics.org

Received: 14 June 2013, Accepted: 9 July 2013
Edited by: G. B. Mindlin
Licence: Creative Commons Attribution 3.0
DOI: http://dx.doi.org/10.4279/PIP.050006

ISSN 1852-4249

arXiv:1312.6660v1 [q-bio.NC] 10 Dec 2013

# A neuronal device for the control of multi-step computations

Ariel Zylberberg,[1–4*] Luciano Paz,[1,2†] Pieter R. Roelfsema,[4–6‡] Stanislas Dehaene,[7–10§] Mariano Sigman[2¶]

We describe the operation of a neuronal device which embodies the computational principles of the "paper-and-pencil" machine envisioned by Alan Turing. The network is based on principles of cortical organization. We develop a plausible solution to implement pointers and investigate how neuronal circuits may instantiate the basic operations involved in assigning a value to a variable (*i.e.*, $x = 5$), in determining whether two variables have the same value and in retrieving the value of a given variable to be accessible to other nodes of the network. We exemplify the collective function of the network in simplified arithmetic and problem solving (blocks-world) tasks.

*E-mail: arielz@df.uba.ar
†E-mail: lpaz@df.uba.ar
‡E-mail: p.roelfsema@nin.knaw.nl
§E-mail: stanislas.dehaene@gmail.com
¶E-mail: sigman@df.uba.ar

[1] These authors contributed equally to this work

[2] Laboratory of Integrative Neuroscience, Physics Department, FCEyN UBA and IFIBA, CONICET; Pabellón 1, Ciudad Universitaria, 1428 Buenos Aires, Argentina.

[3] Laboratory of Applied Artificial Intelligence, Computer Science Department, FCEyN UBA; Pabellón 1, Ciudad Universitaria, 1428 Buenos Aires, Argentina.

[4] Department of Vision and Cognition, Netherlands Institute for Neuroscience, an institute of the Royal Netherlands Academy of Arts and Sciences, Meibergdreef 47, 1105 BA Amsterdam, The Netherlands.

[5] Department of Integrative Neurophysiology, Center for Neurogenomics and Cognitive Research, VU University, Amsterdam, The Netherlands.

[6] Psychiatry Department, Academic Medical Center, Amsterdam, The Netherlands.

[7] Collge de France, F-75005 Paris, France.

[8] INSERM, U992, Cognitive Neuroimaging Unit, F-91191 Gif/Yvette, France.

[9] CEA, DSV/I2BM, NeuroSpin Center, F-91191 Gif/Yvette, France.

[10] University Paris-Sud, Cognitive Neuroimaging Unit, F-91191 Gif/Yvette, France.

## I. Introduction

Consider the task of finding a route in a map. You are likely to start searching the initial and final destinations, identifying possible routes between them, and then selecting the one you think is shorter or more appropriate. This simple example highlights how almost any task we perform is organized in a sequence of processes involving operations which we identify as "atomic" (here search, memory and decision-making). In contrast with the thorough knowledge of the neurophysiology underlying these atomic operations [1–3], neuroscience is only starting to shed light on how they organize into programs [4–6]. Partly due to the difficulty of implementing compound tasks in animal models, sequential decision-making has mostly been addressed by the domains of artificial intelligence, cognitive science and psychology [7, 8].

Our goal is to go beyond the available neurophysiological data to show how the brain might sequentialize operations to conform multi-step cognitive

programs. We suppose the existence of elementary operations akin to Ullman's [9] routines, although not limited to the visual domain. Of special relevance to our report is the body of work that has grown out of the seminal work of John Anderson [8]. Building on the notion of "production systems" [10], Anderson and colleagues developed ACT-R as a framework for human cognition [8]. It consists of a general mechanism for selecting productions fueled by sensory, motor, goal and memory modules. The ACT-R framework emphasizes the chained nature of human cognition: at any moment in the execution of a task, information placed in buffers acts as data for the central production system, which feeds-back to these same modules.

Despite vast recent progress in our understanding of decision formation in simple perceptual tasks [3], it remains unresolved how the operations required by cognitive architectures may be implemented at the level of single neurons. We address some of the challenges posed by the translation of cognitive architectures to neurons: how neuronal circuits might implement a single operation, how multiple operations are arranged in a sequence, how the output of one operation is made available to the next one.

## II. Fundamental assumptions and neuronal implementation

### i. The basis for single operations

Insights into the machinery for simple sensory-motor associations come from studies of monkey electrophysiology. Studies of oculomotor decisions —focused primarily on area LIP [12]— have shown that neurons in this area reflect the accumulation of evidence leading to a decision [3]. In a well studied paradigm, monkeys were trained to discriminate the direction of motion of a patch of moving dots and report the direction of motion with an eye-movement response to the target located in the direction of motion [13]. Neurons in LIP which respond with high levels of activity during memory saccades to specific portions of space are recorded during the task. These neurons show ramping activity of their firing rates with a slope that depends on the difficulty of the task, controlled by the proportion of dots that move coherently in one direction [14]. In a reaction time version of the task [13], when monkeys are free to make a saccade any time after the onset of the motion stimuli, the ramping activity continues until a fixed level of activity is reached, signaling the impending saccade (Fig. 1 a). Crucially, the level of this "threshold" does not depend on the difficulty of the task or the time to respond. The emerging picture is that ramping neurons in LIP integrate sensory evidence and trigger a response when activity reaches a threshold. This finding provided strong support for accumulation or race models of decision making which have been previously postulated to explain error rates and reaction times in simple tasks, and match nicely with decision theoretical notions of optimality [15]. While experimental studies have mainly characterized the feed-forward flow of information from sensory to motor areas, evidence accumulation is also modulated by contextual and task-related information including prior knowledge about the likelihood and payoff of each alternative [16, 17]. Interestingly, a common currency —the spiking activity of neurons in motor intention areas— may underlie these seemingly unequal influences on decision formation.

### ii. Sequencing of multiple operations

Brain circuits can integrate sensory evidence over hundreds of milliseconds. This illustrates how the brain decides based on unreliable evidence, averaging over time to clean up the noise. Yet the duration of single accumulation processes is constrained by its analog character, a problem pointed out earlier by von Neumann in his book *The computer and the Brain* [18]: "in the course of long calculations not only do errors add up but also those committed early in the calculation are amplified by the latter parts of it...". Modern computers avoid the problem of noise amplification employing digital instead of analog computation. We have suggested that the brain may deal with the amplification of noise by serially chaining individual integration stages, where the changes made by one ramp-to-threshold process represent the initial stage for the next one (Fig. 1 b) [11, 19].

Evidence for the ramp-to-threshold dynamics has been derived from tasks in which the decision reflects a commitment to a motor plan [3, 20]. As others [21], we posit that the ramp-to-threshold process for action-selection is not restricted to motor actions, but may also be a mechanism to select

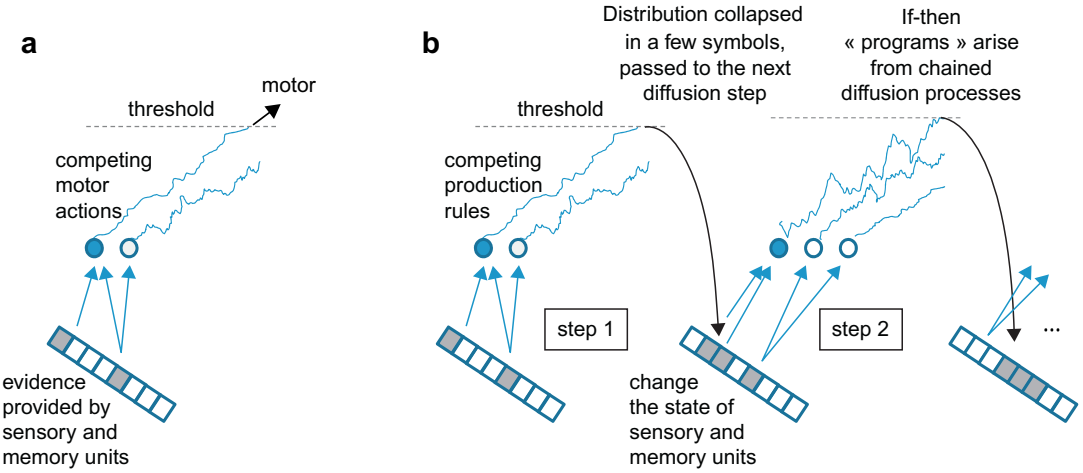



Figure 1: Diffusion processes in single and multi-step tasks. (a) In simple sensory-motor tasks, response selection is mediated by the the accumulation of sensory evidence to a response threshold. (b) In tasks involving multiple steps, there is a parallel competition between a subset of "productions rules" implemented by pools of neurons that accumulate evidence up to a threshold. The selected production can have overt effects (motor actions) as well as covert effects in modifying the state of the memory, after which a new cycle begins. Adapted with permission from Ref. [11].

internal actions like the decision of where to attend, what to store in memory, or what question to ask next. Therefore, the activation of a circuit based on sensory and mnemonic evidence is mediated by the accumulation of evidence in areas of the brain which can activate that specific circuit, and which compete with other internal and external actions. Within a single step, the computation proceeds in parallel and settles in a choice. Seriality is the consequence of the competitive selection of internal and external actions that transforms noisy and parallel steps of evidence accumulation into a sequence of discrete changes in the state of the network. These discrete steps clean up the noise and enable the logical flow of the computation.

Following the terminology of symbolic cognitive architectures [7, 8], the "ramping" neurons which select the operation to do next are referred as "production" neurons. The competition between productions is driven by inputs from sensory and memory neurons and by the spontaneous activity in the production network. As in single-step decisions [13], the race between productions concludes when neurons encoding one production reach a decision threshold. The neurons which detect the crossing of a threshold by a production also mediate its effects, which is to activate or deactivate other brain circuits. The activated circuits can be motor, but are not restricted to it, producing different effects like changes in the state of working memory (deciding what to remember), activating and broadcasting information which was in a "latent" state (like sensory traces and synaptic memories [22,23]), or activating peripheral processors capable of performing specific functions (like changing the focus of attention, segregating a figure from its background, or tracing a curve).

### iii. Pointers

Versatility and flexibility are shared computational virtues of the human brain and of the Turing machine. The simple example of addition (at the root of Turing's conception) well illustrates what sort of architecture this requires. One can picture the addition of two numbers x and y as displacing y steps from the initial position x. This simple representation of addition as a walk in the number line describes the core connection between movement in space and mathematical operations. It also describes the need of operations that use variables which temporarily bind to values in order to achieve flexibility.

In this section we describe how neuronal circuits

may instantiate the basic operations involved in assigning a value to a variable (*i.e.*, $x = 5$), in determining whether two variables have the same value and in retrieving the value of a given variable. This is in a way a proof of concept, *i.e.*, a way to construct these operations with neurons. We are of course in no position to claim that this instantiation is unique (it is certainly not). However, we have tried to ground it on important principles of neurophsyiology and we believe that this construction raises important aspects and limitations which may generalize to other neuronal constructions of variable assignment, comparison and retrieval mechanisms.

Here we introduce the concept of pointers; individual or pool of neurons which can temporarily point to other circuits. When a pointer is active, it facilitates the activation of the circuit to which it is temporarily bound and which is dynamically set during the course of a computation.

A pointer "points" to a cortical territory (for instance, to V1). This cortical mantle represents a space of values that a given variable may assume. The cortex is organized in spatial maps representing features (space, orientation, color, 1-D line...) and a pointer can temporarily bind to one of these possible set of values in a way that the activation of the pointer corresponds to the activation of the value and hence functions as a variable (*i.e.*, $x = 3$). There are many proposed physiological mechanisms to temporally bind neuronal circuits [24–26]. A broad class of mechanisms relies on sustained reverberant neuronal activity [26]. A different class relies on small modifications of synaptic activity, which constitute silent memories in latent connections [23]. Here we opt for the second alternative, first because it has a great metabolic advantage allowing to share many memories at very low cost, and more importantly because it separates the processes of variable assignation and variable retrieval. As we describe below in detail, in this architecture the current state of the variable is not broadcast to other areas until it is specifically queried.

To specifically implement the binding with neuron-like elements, we follow the classic assumption that when two groups of neurons are active at the same time, a subset of the connection between them is strengthened (Fig. 2). The strengthening of the synapses is bidirectional, and it is responsible for the binding between neuronal populations. To avoid saturation of the connections and to allow for new associations to form, the strength of these connections decays exponentially within a few hundred milliseconds. Specifically, if the connection strength between a pair of populations is $w_{base}$, then when both populations are active the connection strength increases exponentially with a time constant $\tau_{rise}$ to a maximum connection strength of $w_{max}$. When one or both of the populations become inactive then the connection strength decays exponentially back to $w_{base}$ with a time constant of $\tau_{decay}$.

The mechanism described above generates a silent coupling between a pointer and the value to which it points. How is this value recovered? In other words, how can other elements of the program know the value of the variable $X$? The expression of a value stored in silent synapses is achieved by simultaneously activating the pointer circuit and forcing the domain of the variable to a winner-take-all (WTA) mode (Fig. 2). The WTA mode —set by having neurons with self-excitation and cross-inhibition [27]— assures that only one value is retrieved at a time. These neurons make stronger connections with the neurons to which they are bound than with the other neurons. When the network is set in a WTA mode, these connections bias the competition to retrieve the value previously associated with the pointer (Fig. 2). In other words, activation of the pointer by itself is not sufficient to drive synaptic activity to the neuron (or neurons) representing the value to which it points. But it can bias the activation of a specific neuron when co-activated with a tonic activation of the entire network.

This architecture is flexible and economic. Value neurons only fire when they are set or retrieved. Memory capacity is constrained by the number of connections and not by the number of neurons. But it also has a caveat. Given that only one variable can be bound to a specific domain at any one time, multiple bindings must be addressed serially. As we show later, this can be accomplished by the sequential firing of production neurons.

### *a. Compare the value of two variables*

If two variables $X$ and $Y$ bind to instances in the same domain, it is possible to determine whether the two variables are bound to the same instance,

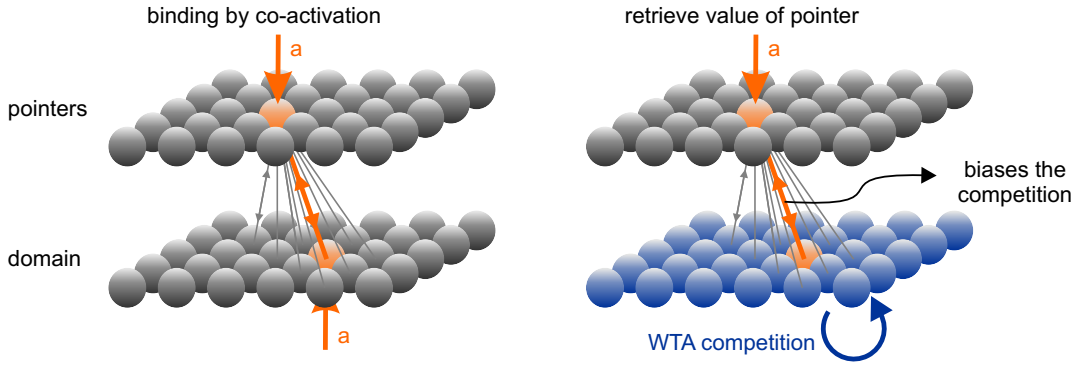



Figure 2: Instantiation of variables through plastic synapses. When two neurons become active, the connection between them is rapidly strengthened, forming a transient association which decays within a few hundreds of milliseconds. The value of a pointer can be retrieved by setting the pointer's domain in a winner-take-all mode and activating the pointer which biases the WTA competition through its strengthened connections.

*i.e.*, whether $x = y$. The mechanics of this process is very similar to retrieving the value of a variable.

Pointer neurons[1] $X$ and $Y$ are co-activated. The equality in the assignment of $X$ and $Y$ can be identified by a coincidence detector. Specifically, this is solved by adjusting the excitability in the value domain in such a way that the simultaneous input on a single value neuron exceeds the threshold but the input of a single pointer does not. This proposed mechanism is very similar to the circuits in the brain stem which —based on coincidence detection of delay lines— encode interaural time difference [28]. This shows the concrete plausibility of generating such dynamic threshold mechanisms that act as coincidence detectors.

### *b. Assign the value of one variable to another*

Similarly, to assign the value of $X$ to the variable $Y$ ($Y \leftarrow x$), the value of the variable that is to be copied needs to be retrieved as indicated previously, by activating the variable $X$ and forcing a WTA competition at the variable's domain. Then, the node coding for variable $Y$ must be activated, which will lead to a reinforcing of the connections between $Y$ and $x$ which will instantiate the new association.

[1] In our framework, a pointer can also be a population of neurons that functions as a single pointer.

## III. Concrete implementation of neuronal programs

In the previous sections we sketched a set of principles by which brain circuits can control multi-step operations and store temporary information in memory buffers to share it between different operations.

Here we demonstrate, as a proof of concept, a neuronal implementation of such circuits in two simple tasks.

The first one is a simple arithmetic counter, where the network has to count from an initial number $n_{ini}$ to a final number $n_{end}$, a task that can be seen as the emblematic operation of a Turing device. The second example is a blocks-world problem, a paradigmatic example of multi-step decision making in artificial intelligence [29]. The aim of the first task is to illustrate how the different elements sketched above act in concert to implement neuronal programs. The motivation to implement blocks worlds is to link these ideas to developed notions of visual routines [9, 30, 31].

### i. Arithmetic counter

We designed the network to be generic in the sense of being able to solve any instance of the problem, *i.e.*, any instantiation of $n_{ini}$ and $n_{end}$. We decided to implement a counter, since it constitutes essentially a while loop and hence a basic intermediate description of most flexible computations. In the

network, each node is meaningful, and all parameters were set by hand. Of course, understanding how these parameters are adjusted through a learning process is a difficult and important question, but this is left for future work.

Each number is represented by a pool of neurons selective to the corresponding numerosity value [32]. A potential area for the neurons belonging to the numerosity domain is the Intra Parietal Sulcus (IPS) [33], where neurons coding for numerosity have been found in both humans and monkeys [34]. In the model, number neurons interact through random lateral inhibitory connections and self-excitation. This allows, as described above, to collapse a broad distribution of number neurons [32] to a pool representing a single number, in a retrieval process during a step of the program. We assume that the newtwork has learned a notion of number proximity and continuity. This was implemented via a transition-network that has asymmetrical connections with the number-network. A given neuron representing the number $n$ excites the transition neuron $n \to n+1$ population. This in turn excites the neuron that represents the number $n+1$. Again, we do not delve into how this is learned, we assume it as a consolidated mechanism.

The numbers-network can be in different modes: it can be quiescent, such that no number is active, or it can be in a winner-take-all mode with only one unit in the active state. Our network implementation of the counter makes use of two variables. The *Count* variable stores the current count and changes dynamically as the program progresses, after being initialized to $n_{ini}$. The *End* variable stores the number at which the counting has to stop and is initially set to $n_{end}$.

The network behaves basically as a while loop, increasing the value of the *Count* variable while its value differs from that of the *End* variable. To increment the count, we modeled a transition-network with units that have asymmetrical connections with the numbers-network. For example, the "$1 \to 2$" node receives excitatory input from the unit coding for number 1 and in turn excites number 2. This network stores knowledge about successor functions, and in order to become active it requires additional input from the production system. As mentioned above, here we do not address how such structure is learned in number representing neurons. Learning to count is an effortful process during childhood [35] by which children learn transition rules between numbers. We postulate that these relations are encoded in structures which resemble horizontal connections in the cortex [36–38]. In the same way that horizontal connections incorporate transition probabilities of oriented elements in a slow learning process [39, 40], resulting in a Gestalt as a psychological sense of "natural" continuity, we argue that horizontal connections between numerosity neurons can endow the same sense of transition probability and natural continuity in the space of numbers. The successor function can be as an homologous to a matrix of horizontal connections in the array of number neurons.

In a way, our description postulates that a certain number of operations are embedded in each domain cortex (orientation selective neurons in V1 for curve tracing, number selective neurons in IPS for arithmetic...). This can be seen as "compiled" routines which are instantiated by local horizontal connections capable of performing operations such as collinear continuity, or "add one". The program can control which of these operations becomes active at any given step by gating the set of horizontal connections, a process we have referred to as "addressing" the cortex [41]. Just as an example, when older children learn to automatically count every three numbers (1, 4, 7, 10, 13...) we postulate that they have instantiated a new routine (through a slow learning process) capable of establishing the transition matrix of $n \to n+3$. The repertoire of compiled functions is dynamic and can change with learning [42].

Counting requires a sequence of operations which include changes in the current count, retrieval of successor functions and numeric comparisons. The successive steps of the counting routine are governed by firing of production neurons. The order in which the productions fire is controlled by the content of the memory (Fig. 3). We emphasize that while the production selection process proceeds in parallel —as each production neuron constantly evaluates its input— the selected production strongly inhibits the other production neurons and therefore the evidence accumulated at one step is for the most part lost after a production is selected.[2] In Fig. 3 we simulate a network that has

[2] In the absence of external noise (as in the present simulations), only the production with the largest input has

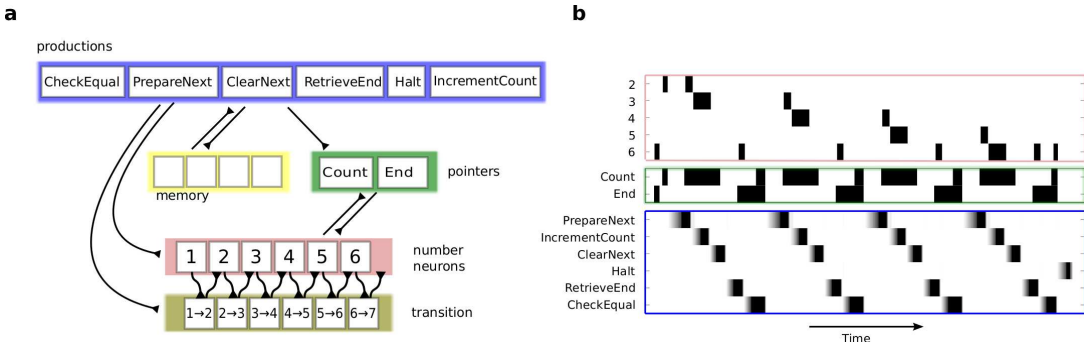



Figure 3: Sketch of the network implementing an arithmetic counter. (a) The network is divided into five sub-networks: productions, memory, pointers (or variables), numbers, and transition-networks. The order in which the productions fire is controlled by the state of the memory network, which is itself controlled by the production system. (b) Dynamics of a subset of neurons in the network. All units are binary except for the production neurons (violet) which gradually accumulate evidence to a threshold.

to count from numbers 2 to 6. Once the initial and final numbers have been bound to the *Count* and *End* variables respectively, the network cycles through six productions. The first production that is selected is the *PrepareNext* production, whose role is to retrieve the value that results from adding 1 to the current count. To this end, this production retrieves the current value of the *Count* variable, and excites the neurons of the transition-network such that the node receiving an input from the retrieved value of *Count* becomes active (*i.e.*, if 2 is active in the numbers-network, then $2 \rightarrow 3$ becomes active in the transition-network). To assure that the retrieved value is remembered for the next step (the actual change of the current count), neurons in the transition-network are endowed with recurrent excitation, and therefore these neurons remain active until explicitly inhibited. The same production also activates a node in the memory-network which excites the *IncrementCount* production, which is therefore selected next. The role of the next production (*IncrementCount*) is to actually update the current count, changing the binding of the *Count* variable. The *IncrementCount* production inhibits all neurons in the number network, to turn it to the quiescent state. Once the network is quiescent, lateral inhibition between number nodes is released and the asymmetrical inputs from the transition-network can activate the number to which it projects. At the same time, the *IncrementCount* production activates the *Count* variable, which is then bound to the currently active node in the numbers-network. Notice that this two-step process between the *PrepareNext* and *IncrementCount* productions basically re-assigns the value of the *Count* variable from its initial value $n$ to a new value $n + 1$. Using a single production to replace these two (as tried in earlier versions of the simulations) required activating the number and transition neurons at the same time which lead to fast and uncontrolled transitions in the numbers-network. To increase the current count in a controlled manner, we settled for the two-productions solution. The *IncrementCount* production also activates a memory unit that biases the competition at the next stage, in favor of the *ClearNext* production. This production shuts up the activity in the transition-network, strongly inhibiting its neurons to compensate for their recurrent excitation. Shutting the activity of these neurons is required at the next step of the routine to avoid changes in the current count when the *Count* variable is retrieved to be compared with the *End* variable. After *ClearNext*, the *RetrieveEnd* production fires which retrieves the value of the *End* variable to strengthen the connections between the *End* variable and the value to which it is bound. This step is required since the strength of the plastic connections decays rapidly, and therefore the instantiation of the variables will be lost if not used or reactivated periodically. Finally, the *CheckEqual* production is selected to determine if the *Count* and *End* variables are equal. If both variables are equal, a node

higher-than-baseline activity.

in the memory network is activated which is detected by the *Halt* production to indicate that the task has been completed; otherwise, the production that is selected next is the *PrepareNext* production and the production cycle is repeated. In Fig. 3b, we show the dynamics of a subset of neurons for a network that has to count from 2 to 6.

With this example we have shown how even a seemingly very easy task such as counting (which can be encoded in up to two lines in virtually any programming language) seems to require a complex set of procedures to coordinate and stabilize all computations, when they are performed by neuronal circuits with slow building of activity and temporal decay.

### ii. A world of blocks

A natural extension of the numerotopic domain used in the above example is to incorporate problems in which the actor must interact with its environment, and sensory and motor productions ought to be coordinated.

The visual system performs a great variety of computations. It can encode a large set of visual features in a parallel feed-forward manner forming its base representation [9,30,43,44], and temporally store these features in a distributed manner [45–47]. A matrix of lateral connections gated by top-down processing can further detect conjunctions of these feature for object recognition.

In analogy with motor routines, the visual system relies on serial chaining of elemental operation [9, 31] to gain computational flexibility. There are many proposals as to which operations are elemental [31], but, as we have discussed above, this list may be fuzzy since the set of elementary operations may be changed by learning [41]. In this framework, atomic operations are those that are encoded in value domains.

Here, and for the purpose of implementing a neural circuit capable of solving the blocks worlds, we will focus on a simplified group of three elemental operations:

**Visual search:** the capacity of the system to identify the location of a given feature.

**Visual cuing:** the capacity of the system of highlighting the features that are present at a given location.

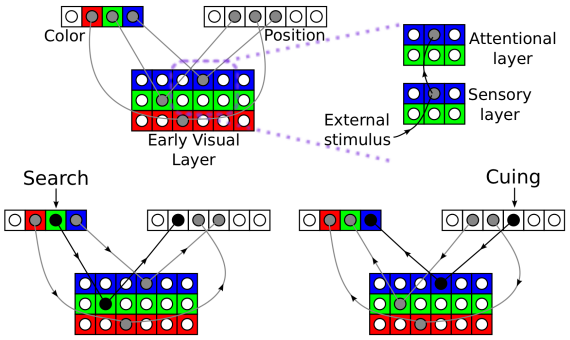



Figure 4: Simplified model of the visual system used in the blocks-world simulations. The upper portion shows the different layers and their connections. The early visual area is formed by a first sensory layer of neurons that receive stimulation from the outer world and a second attentional layer with bidirectional connections between the higher color or position tuned areas. The grayed neurons are the ones that present a higher activity. The lower panels show an example of the cuing and search operations. In the latter, a color tuned neuron is stimulated and drives an activity increase in the early visual layer. That later empowers the activity in the position layer concluding the search. The right situation shows the similar cuing operation.

**Shift the processing focus:** a method guided by attention to focus the processing of visual features or other computations in a given location.

Here we use a simplified representation of the visual system based on previous studies [48–50]. We assume a hierarchy of two layers of neurons. The first one is tuned to conjunctions of colors and locations in the visual field. The second one has two distinct groups of populations, one with neurons that have large receptive fields that encode color irrespective of their location and another which encodes location independently of color (Fig. 4). The model assumes that neurons in the first layer tuned to a particular color and retinotopic location are reciprocally connected to second layer units that encode the same color or location.

This architecture performs visual search of color in a way which resembles the variable assignation described above, through a conjunction mechanism between maps encoding different features. In the

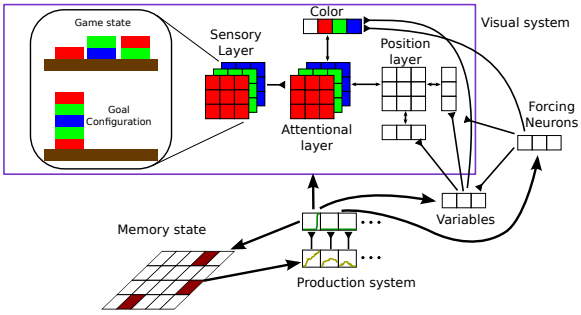



Figure 5: Sketch of the network that solves the blocks-world problem. The sensory early visual system receives input from the BW configuration and excites the first attentional layer. The latter is connected to color and position specific areas. The arrows show there is a connection between layers. The individual connections may be excitatory or inhibitory. The connections with the inverted triangle head indicate only excitatory connections exist.

model, the color cortex encodes each color redundantly for all positions forming a set of spatial vectors (one for each color). Of course, all these spatial maps selective for a given color can be intermingled in the cortex, as it is also the case with orientation columns which sample all orientations filling space in small receptive fields. If, for instance, a red square is presented in position three, the neuron selective for red (henceforth referred to as in the red map) and with a receptive field in position three will fire. This activation, in turn, propagates to spatial neurons (which are insensitive to color). Thus, if four squares of different colors are presented in positions 1 to 4, the spatial neurons in these positions will fire at the same rate. To search for the spatial position of a red block, the activity of neurons coding for red in the color map must be enhanced. The enhanced activity propagates to the early visual areas which code for conjunctions of color and space, which in turn propagates to the spatial map, highlighting the position where the searched color was present. Spatial selection is triggered by an attention layer which selects the production "attend to red", addressing the sensory cortex in a way that only locations containing red features will be routed to the spatial neurons. The color of a block at one location can be retrieved by an almost identical mechanism. In this case, the production system sets the attentional network to a given position in space and through conjunction mechanisms (because connections are reciprocal) only the color in the selected position is retrieved. This is a simple device for searching based on the propagation of attentional signals which has been used before in several models (*e.g.*, Ref. [31]) (Fig. 4).

To bridge these ideas which are well grounded in the visual literature [30, 31, 43] with notions of planning and sequential mental programs, we use this model to implement a solver for a simple set of Blocks-World problems. The blocks-world framework is a paradigmatic artificial intelligence problem that consists of a series of colored blocks piled up on top of a large surface in many columns. The goal is to arrange the blocks according to their color in a given goal configuration. The player can only move one block from the top of any column and place it at the top of another, or on the surface that supports all the blocks. We choose to construct a solver for a restricted blocks-world problem where the surface can only hold 3 columns of blocks and the goal configuration is to arrange them all into one column (that we call the target column).[3]

We implement a network with a set of memory and production neurons —analogous to the counter circuit described above— which coordinates a set of visual and motor productions (Fig. 5). The interaction between the memory layer and the production system triggers the execution of elemental visual processes, motor actions and changes in the memory configuration in order to solve any given instance of the problem.

To solve this problem, an algorithm needs to be able to find whether a block is in the correct position. For this, it requires, first, a "retrieve color" from a given location function. Normally the location that is intended to be cued is the one that is being attended to. We implement the attended location as a variable population (that we call the processing focus or $PF$ inspired in Ullman's work [9]), so the "retrieve color" is equal to cue the color in $PF$'s location.[4] Second, it must compare the colors in different locations. This can be done by bind-

[3] This restricted problem is equivalent to the Tower of London game [51].

[4] There are works that name $PF$ as *Deictic Pointers* and suggest that it would be possible to store it also by keeping gaze or even a finger at the relevant location [52].

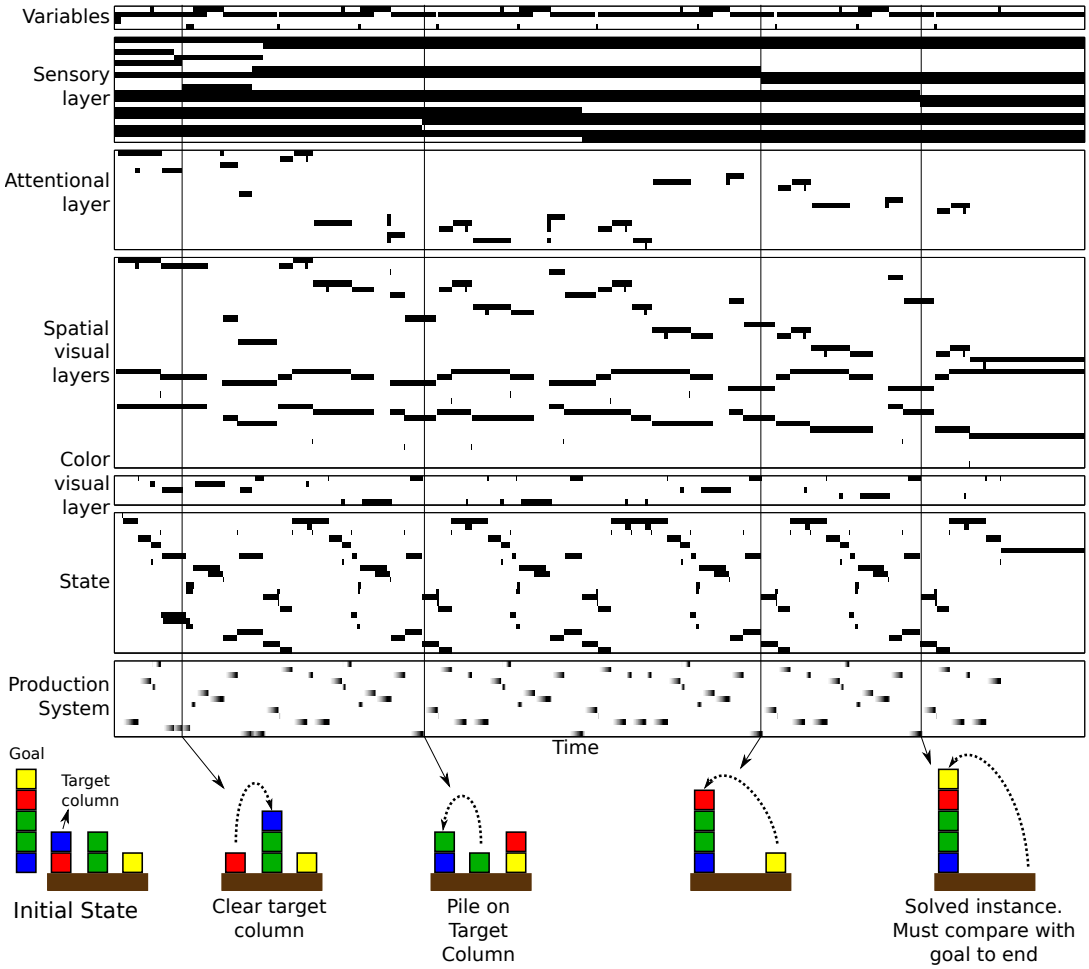



Figure 6: Mean firing rate of a subset of neuronal populations involved in the resolution of an instance of BW. The rate is normalized between 0 (white) and 1 (black). The horizontal axis represents the ellapsed time in arbitrary units. At the bottom, we show a subset of intermediate BW configurations, aligned to the execution time of the motor commands which lead to these configurations.

ing the relevant location colors to separate variables and then comparing them in the way described in section a.

As the goal is to pile all the blocks in the correct order in a given target column, a possible first step towards the goal is to compare the target column with the goal configuration from the bottom to the top. This can be done by chaining several movements of the processing focus with color retrievals and subsequent comparisons. Once the first difference is found, the target column's upper blocks must be moved away in order to replace the different colored block with the correct one. This process is carried out using several motor productions. Once the target column is free to receive the correct colored block, that color must be searched in the remaining columns. This is done as described earlier in this section. Once found, the $PF$ can be moved there in order to view if there are blocks above it. If there are, motor productions must be chained in order to free the desired block and move it to the target column. After this is done, the program can loop back to comparing the target column with the goal configuration and iteratively solve the problem.

Our neuronal implementation chains the produc-

tions in a similar way as the one described above and elicits a complex activity pattern (Fig. 6). A detailed explanation of the implementation can be found in the supplementary material [53].

## IV. Conclusions

Here we presented ideas aimed to bridge the gap between the neurophysiology of simple decisions and the control of sequential operations. Our framework proposes a specific set of mechanisms by which multi-step computations can be controlled by neural circuits. Action selection is determined by a parallel competition amongst competing neurons which slowly accumulate sensory and mnemonic evidence until a threshold. Actions are conceived in a broad sense, as they can result in the activation of motor circuits or other brain circuits not directly involved in a movement of the eyes or the limbs. Thresholding the action of the productions results in discrete changes to a meta-stable network. These discrete steps clean up the noise and enable a logical flow of the computation.

Comprehending the electrophysiological mechanisms of seriality is hindered by the intrinsic difficulty of training non-human primates in complex multi-step tasks. The ideas presented in this report may serve to guide the experimental search for the mechanisms required to perform tasks involving multiple steps. Neurons integrating evidence towards a threshold should be observed even in the absence of an overt response, for instance in the frontal-eye fields of awake monkeys for the control of attention. Memory neurons should show fast transitions between metastable states, on average every $\sim$ 100-250 msec, compatible with the mean time between successive productions in ACT-R [8].

As mentioned, we do not address how the productions and the order in which they are executed are learned. There is a vast literature, for instance in reinforcement learning [54–56] describing how to learn the sequence of actions required to solve a task. Deahene & Changeaux [57] showed how a neuronal network can solve a task similar to the BW that we modelled here, but where the order in which productions fire was controlled by the distance from the game state to the goal. Instead, our aim here was to investigate how the algorithm (the pseudo-code) may be implemented in neuronal circuits —once it has already been learned— from a small set of generic principles. The operation of the proposed neuronal device in a simple arithmetic task and in a neuronal network capable of solving any instance of a restricted Blocks-World domain illustrates the plausibility of our framework for the control of computations involving multiple steps.

*Acknowledgements* - AZ was supported by a fellowship of the Peruilh Foundation, Faculty of Engineering, Universidad de Buenos Aires. LP was supported by a fellowship of the National Research Council of Argentina (CONICET). PRR was supported by a Netherlands Organization for Scientific Research (NWO)-VICI grant, a grant of the NWO Excellence Program Brain and Cognition, and a Human Frontier Science Program grant.

[1] M Platt, P Glimcher, *Neural correlates of decision variables in parietal cortex*, Nature **400**, 233 (1999).

[2] J D Schall, *Neural basis of deciding, choosing and acting*, Nat. Rev. Neurosci. **2**, 33 (2001).

[3] J I Gold, M N Shadlen, *The neural basis of decision making*, Annu. Rev. Neurosci. **30**, 535 (2007).

[4] P R Roelfsema, P S Khayat, H Spekreijse, *Subtask sequencing in the primary visual cortex.*, P. Natl. Acad. Sci. USA **100**, 5467 (2003).

[5] R Romo, E Salinas, *Flutter discrimination: Neural codes, perception, memory and decision making*, Nat. Rev. Neurosci. **4**, 203 (2003).

[6] S I Moro, M Tolboom, P S Khayat, P R Roelfsema, *Neuronal activity in the visual cortex reveals the temporal order of cognitive operations*, J. Neurosci. **30**, 16293 (2010).

[7] A Newell, *Unified theories of cognition*, Harvard University Press, Cambridge, Massachusetts (1990).

[8] J R Anderson, C J Lebiere, *The atomic components of thought*, Lawrence Erlbaum, Mahwah, New Jersey (1998).

[9] S Ullman, *Visual routines*, Cognition **18**, 97 (1984).

[10] A Newell, *Productions systems: Models of control structures*, In: Visual Information Processing, Ed. W G Chase, Pag. 463, Academic Press, New York (1973).

[11] S Dehaene, M Sigman, *From a single decision to a multi-step algorithm*, Curr. Opin. Neurobio. **22**, 937 (2012).

[12] J Gottlieb, P Balan, *Attention as a decision in information space*, Trends Cogn. Sci. **14**, 240 (2010).

[13] J D Roitman, M N Shadlen, *Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task*, J. Neurosci. **22**, 9475 (2002).

[14] M N Shadlen, W T Newsome, *Motion perception: Seeing and deciding*, P. Natl. Acad. Sci. USA **93**, 628 (1996).

[15] Y Huang, A Friesen, T Hanks, M Shadlen, R Rao, *How prior probability influences decision making: A unifying probabilistic model*, In: Advances in Neural Information Processing Systems 25, Eds. P Bartlett, F C N Pereira, C J Ca L Burges, L Bottou, K Q Weinberger, Pag. 1277, Lake Tahoe, Nevada (2012).

[16] L P Sugrue, G S Corrado, W T Newsome, *Matching behavior and the representation of value in the parietal cortex*, Science **304**, 1782 (2004).

[17] J D Wallis, K C Anderson, E K Miller, *Single neurons in prefrontal cortex encode abstract rules*, Nature **411**, 953 (2001).

[18] J Von Neumann, *The computer and the brain*, Yale University Press, New Haven, Connecticut (1958).

[19] A Zylberberg, S Dehaene, P R Roelfsema, M Sigman, *The human Turing machine: A neural framework for mental programs*, Trends Cogn. Sci. **15**, 293 (2011).

[20] G Maimon, J A Assad, *A cognitive signal for the proactive timing of action in macaque lip*, Nat. Neurosci. **9**, 948 (2006).

[21] M N Shadlen, R Kiani, T D Hanks, A K Churchland, *Neurobiology of decision making an intentional framework*, In: Better Than Conscious?, Eds. C Engel, W Singer, Pag. 71, MIT Press, Massachusetts (2008).

[22] A Zylberberg, S Dehaene, G B Mindlin, M Sigman, *Neurophysiological bases of exponential sensory decay and top-down memory retrieval: A model*, Front. Comput. Neurosci. **3**, 4 (2009).

[23] G Mongillo, O Barak, M Tsodyks, *Synaptic theory of working memory*, Science **319**, 1543 (2008).

[24] R C O'Reilly, *Biologically based computational models of high-level cognition*, Science **314**, 91 (2006).

[25] L Shastri, V Ajjanagadde, *et al.*, *From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony*, Behav. Brain Sci. **16**, 417 (1993).

[26] R Hahnloser, R J Douglas, M Mahowald, K Hepp, *Feedback interactions between neuronal pointers and maps for attentional processing*, Nat. Neurosci. **2**, 746 (1999).

[27] X j Wang, *Introduction to computational neuroscience*, Technical report Volen Center for Complex Systems, Brandeis University, Waltham, Massachusetts (2006).

[28] C Carr, M Konishi, *A circuit for detection of interaural time differences in the brain stem of the barn owl*, J. Neurosci. **10**, 3227 (1990).

[29] J Slaney, S Thiébaux, *Blocks world revisited*, Artif. Intell. **125**, 119 (2001).

[30] P R Roelfsema, V A Lamme, H Spekreijse, *The implementation of visual routines*, Vision Res. **40**, 1385 (2000).

[31] P R Roelfsema, *Elemental operations in vision*, Trends Cogn. Sci. **9**, 226 (2005).

[32] S Dehaene, J P Changeux, *Development of elementary numerical abilities: A neuronal model*, J. Cognitive Neurosci. **5**, 390 (1993).

[33] M Piazza, V Izard, P Pinel, D Le Bihan, S Dehaene, *Tuning curves for approximate numerosity in the human intraparietal sulcus*, Neuron **44**, 547 (2004).

[34] A Nieder, S Dehaene, *Representation of number in the brain*, Annu. Rev. Neurosci. **32**, 185 (2009).

[35] C Lebiere, *The dynamics of cognition: An ACT-R model of cognitive arithmetic*, Doctoral dissertation. Carnegie Mellon University, Pittsburgh, Pennsylvania (1998).

[36] D Y Ts'o, C D Gilbert, T N Wiesel, *Relationships between horizontal interactions and functional architecture in cat striate cortex as revealed by cross-correlation analysis*, J. Neurosci. **6**, 1160 (1986).

[37] B A McGuire, C D Gilbert, P K Rivlin, T N Wiesel, *Targets of horizontal connections in macaque primary visual cortex*, J. Comp. Neurol. **305**, 370 (1991).

[38] C D Gilbert, Y Daniel, T N Wiesel, *Lateral interactions in visual cortex*, In: From pigments to perception, Eds. A Valberg, B B Lee, Pag. 239, Plenun Press, New York (1991).

[39] M Sigman, G A Cecchi, C D Gilbert, M O Magnasco, *On a common circle: Natural scenes and gestalt rules*, P. Natl. Acad. Sci. USA **98**, 1935 (2001).

[40] C D Gilbert, M Sigman, R E Crist, *The neural basis of perceptual learning*, Neuron **31**, 681 (2001).

[41] C D Gilbert, M Sigman, *Brain states: Top-down influences in sensory processing*, Neuron **54**, 677 (2007).

[42] M K Kapadia, M Ito, C D Gilbert, G Westheimer, *Improvement in visual sensitivity by changes in local context: Parallel studies in human observers and in v1 of alert monkeys*, Neuron **15**, 843 (1995).

[43] V A Lamme, P R Roelfsema, *The distinct modes of vision offered by feedforward and recurrent processing.*, Trends Neurosci. **23**, 571 (2000).

[44] S Thorpe, D Fize, C Marlot, *Speed of processing in the human visual system*, Nature **381**, 520 (1996).

[45] D J Felleman, D C Van Essen, *Distributed hierarchical processing in the primate cerebral cortex*, Cereb. Cortex **1**, 1 (1991).

[46] G Sperling, *The information available in brief visual presentations*, Psychol. Monogr. Gen. A. **74**, 1 (1960).

[47] M Graziano, M Sigman, *The dynamics of sensory buffers: Geometric spatial and experience-dependent shaping of iconic memory*, J. Vision **8**, 1 (2008).

[48] F Hamker, *The role of feedback connections in task-driven visual search*, In: Connectionist models in cognitive neuroscience, Eds. D Heinke *et al.*, Pag. 252, Springer–Verlag, London (1999).

[49] F H Hamker, *A dynamic model of how feature cues guide spatial attention*, Vision Res. **44**, 501 (2004).

[50] D Heinke, G W Humphreys, *Attention, spatial representation, and visual neglect: Simulating emergent attention and spatial memory in the selective attention for identification model (saim).*, Psychol. Rev. **110**, 29 (2003).

[51] T Shallice, *Specific impairments of planning*, Phil. Trans. R. Soc. Lond. B **298**, 199 (1982).

[52] D H Ballard, M M Hayhoe, P K Pook, R P N Rao, *Deictic codes for the embodiment of cognition*, Behav. Brain Sci. **20**, 723 (1997).

[53] A Zylberberg, L Paz, P R Roelfsema, S Dehaene, M Sigman, *Supplementary Material to this paper*, available at *www.papersinphysics.org* (2013).

[54] R S Sutton, A G Barto, *Reinforcement learning: An introduction*, MIT Press, Massachusetts (1998).

[55] P R Roelfsema, A van Ooyen, T Watanabe, *Perceptual learning rules based on reinforcers and attention.*, Trends Cogn. Sci. **14**, 64 (2010).

[56] J O Rombouts, S M Bohte, P R Roelfsema, *Neurally plausible reinforcement learning of working memory tasks*, In: Advances in Neural Information Processing Systems 25, Eds. P Bartlett, F C N Pereira, C J Ca L Burges, L Bottou, K Q Weinberger, Pag. 1880, Lake Tahoe, Nevada (2012).

[57] S Dehaene, J P Changeux, *A hierarchical neuronal network for planning behavior*, P. Natl. Acad. Sci. USA **94**, 13293 (1997).