

Behaviorism Revisited: Linking Perception and Action Through Symbolic Models of Cognition

Pierre Bonzon¹

¹ Dept of Information Systems, Faculty of HEC, University of Lausanne, Switzerland

`pierre.bonzon@unil.ch`

Abstract. Brain simulations as performed today in computational neuroscience rely on analytical methods i.e., mainly differential equations, to model physical processes. The question then arises: will cognitive abilities of real brains spontaneously emerge from these simulations, or should they be encoded and executed on top of them, similarly to the way computer software runs on hardware? Towards this later goal, a new framework linking neural dynamics to behaviors through a virtual machine has been reported and is used here to model brain functionalities in two domains, namely evolutive cases of analogical reasoning and a simple case of meta-cognition. It is argued that this approach to brain modeling could lead to an actualization of the concept of behaviorism as a model for the development of cognition.

Keywords: brain simulation, virtual machine, neural dynamics, cognitive process, analogical reasoning, meta-cognition

1 Introduction

Since the advent of the cognitive revolution [1], the psychological study of behavior has moved away from behaviorist theories based on associative explanations [2] towards the description of mental states and cognitive processes [3]. Early models of cognitive processes (see e.g., Anderson et al 2004 [4]) were symbolic information processors originating from the research in Artificial Intelligence. These *top down* approaches allow for performing behavior simulations regardless of any grounding issues i.e., they do not address the problem of binding cognitive processes to underlying brain processes. In parallel, another field of research focuses on the actual support of cognition i.e., on brain structures. Computational platforms for simulating networks of neurons range from artificial neural networks (McCulloch & Pitts 1943 [5]) to enhanced neural networks simulators based on differential equations (Hodgkin & Huxley 1952 [6]). These *bottom up* approaches detailing the ground level constituents of the brain (i.e., the neuronal cells and their connections) are not explicitly related to the cognitive tasks they support, and as such are not concerned with the definition of processes. Attempts have been made to turn cognitive models into so-called “neurally plausible cognitive architectures” associated with some form

of brain modeling. As an example, *symbolic connectionist models* (see e.g., Shastri & Ajjanagadde 1993 [7]; Hummel & Holyoak 2003 [8]) constitute an attempt to conciliate the benefits of artificial neural networks and the versatility of the symbolic models of Artificial Intelligence. According to a notable attempt in this direction (Jilk et al. 2008 [9]), "the incommensurable categories at the various levels of description will remain necessary to explain the full range of phenomena".

Another recurrent theme is illustrated by the progressive shift from the traditional "sense-think-act" paradigm of cognitivist systems [10] towards the simplified "sense-act" paradigm of *embodied* cognition [11]. According to this new approach, cognitive processes are executed in connection structures that link sensory circuits (i.e., perception) with motors (i.e., action) [12]. In this perspective, bottom-up approaches based on artificial neural networks, as well as top-down approaches via analytical and/or abstract mathematical tools such as Bayesian inference rules, are well suited for describing "computations" according to Marr's classical tri-level hypothesis [13] that distinguishes computational, algorithmic and implementation levels. They fail however to "identify algorithms and underlying circuits" [14]. In other words, the mechanisms for structure learning remain to be fully understood. What is then needed is a "middle-out" approach that can lead to plausible structures linking biology and cognition.

As a possible solution, we recently proposed [15] to resort to computer science methods that allow for a delineation and implementation of successive levels of complexity. Among the concepts that could be applied towards this goal, two are of particular relevance, namely *concurrent communicating systems*, on one hand, and *virtual machines*, on the other. The notion of a concurrent communicating system, which can be used to model the interaction of objects obeying various communication protocols, reflects a high level view of a network of interactive neurons. The concept of a virtual machine interpreting a compiled code that differs from a processor's native code constitutes the key mechanism that allows for interfacing high level abstract objects i.e., software, with their low level physical support i.e., hardware. In the context of a multi-level model of brain structures and processes, this means that low level physiological details can be put aside, and grounded models of cognition be formulated by relating input and output (i.e., perception and behavior) at a *symbolic level*. The corresponding formal framework distinguishes itself from the usual methods by focusing on processes. It relies for this on a fundamental concept of mathematical logic i.e., the formal notions of an *object in context* represented by symbolic expressions in a logical language [16]. In other words, our virtual machine, which does function as an interface between the neural and cognitive levels, actually performs *contextual deductions*.

This formalism will be reviewed in the Material and methods section below. An assessment of brain functionalities in two domains, namely evolutive analogical reasoning, on one hand, and a simple case of meta-cognition, on the other, is proposed in the Results section under the form of virtual circuits. It is argued in the Conclusion that this overall approach to brain modeling could lead to a revival of the old concept of behaviorism as a model for the development of cognition.

2 Material and methods

Our overall methodology can be described as follows (see [15] for more details):

- a) *micro scale* virtual circuits implementing synaptic plasticity through asynchronous communicating processes are first defined
- b) *meso scale* virtual circuits corresponding to basic cognitive processes are then composed out of these micro scale circuits
- c) both types of virtual circuits are finally compiled into *virtual code* to be interpreted by a virtual machine.

The concept of a virtual machine that we use here basically allows for emulating the execution of a program in language S on a system having its own language L . Such a machine constitutes an interface which allows for defining mesoscale circuits independently of the way the microcircuits are actually implemented. Mesoscale circuits thus somehow correspond to cognitive software running on top of a biological substrate. More precisely, these circuits define source code expressed in S which are then compiled into virtual object code expressed in L to be interpreted by the virtual machine. We follow a bidirectional approach and present in turn the *bottom up* design of virtual circuits followed by the *top down* construction of a virtual machine.

2.1 Bottom up design of virtual circuits

The basic entities of the proposed formalism for representing virtual circuits are constituted by *threads*. In computer science, a thread is a sequence of instructions that executes concurrently with other threads, may coexist with other threads to form a process and share resources such as memory. In the present context, a thread corresponds to a single or a group of neurons and will be represented by a symbolic expression enclosing an instruction *tree* (see below for the definition of the corresponding formal language S). Threads are communicating entities. Each communication does involve a pair of threads and entails on one side the signal transmitted by a pre-synaptic (source) thread, and on the other side its reception, via a given synapse, by a post-synaptic (recipient) thread. Similarly to a neuron, a thread can be both a source and a recipient and functions as a gate receiving incoming signals from different sources and sending an outgoing signal to possibly many recipients. There are however two essential differences between threads and neurons that allow for a single thread to represent a group of neurons i.e.,

- contrary to a neuron that alternates roles, a thread can be simultaneously a source and a recipient by maintaining parallel communications
- contrary to traditional neuron models in which incoming signals are summed into an integrated value, thread inputs can be processed individually.

Threads can be grouped into disjoint sets, or *fibers*, to model neural assemblies (Palm 1982 [17]), and discrete weights can be attached to pairs of communicating threads belonging to the same fiber. The interaction of threads within a given fiber obeys various communication protocols. To illustrate this, we reproduce below from [15] circuits modeling two cases of simple animal behaviors.

2.1.1 A case of classical conditioning

As a general evolution principle, organisms devise and use “tricks” for their survival. The ability to evaluate a threat by learning predictive relationships e.g., by associating a noise and the presence of a predator, is an example of such tricks realized by *classical conditioning*. Let us consider the classical conditioning in the defensive siphon and gill withdrawal reflex of *aplysia californica* [18]. In this experiment, a light tactile conditioned stimulus **cs** elicits a weak defensive reflex, and a strong noxious unconditioned stimulus **us** produces a massive withdrawal reflex. After a few pairings of stimuli **cs** and **us**, where **cs** slightly precedes **us**, a stimulus **cs** alone triggers a significantly enhanced withdrawal reflex i.e., the organism has learned a new behavior. This can be represented by virtual circuit given in Fig. 1.

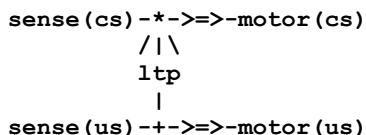


Fig. 1. A mesoscale virtual circuit implementing classical conditioning.

In Figure 1, the threads **sense (us)** and **sense (cs)** are coupled with sensors capturing external stimuli **us** and **cs** and correspond to sensory neurons. The threads **motor (us)** and **motor (cs)** are coupled with effectors and correspond to motor neurons. Finally, the thread **ltp** (for *long term potentiation*) acts as a facilitatory interneuron reinforcing the pathway (i.e. augmenting its *weight*) between **sense (cs)** and **motor (cs)**. The communication protocols depicted by the symbol **->=>-** and **/|** represent a synaptic transmission (i.e., the symbol **->=>-** stands for a *synapse*) and the modulation of a synapse, respectively. The symbols ***** and **+** stand for conjunctive and disjunctive operators i.e., they are used to represent the convergence of incoming signals and the dissemination of an outgoing signal, respectively. Classical conditioning then follows from the application of hebbian learning [19] i.e., “neurons that fire together wire together”. Though it is admitted today that classical conditioning in *aplysia* is mediated by multiple neuronal mechanisms including a postsynaptic retroaction on a presynaptic site, the important issue is that the learning of a new behavior requires a conjoint activity of multiple neurons. This activity in turn depends on the temporal pairing of the conditioned and unconditioned stimuli, which in conclusion leads to implement the thread **ltp** as a *detector of coincidence*.

2.1.2 A simple case of operant conditioning

The ability to assess and to remember the consequences of one's own actions is another example of associative learning providing survival advantages. In this case, operant conditioning [2] associates an action and its result, which can be *positive* or *negative*. Toward this goal, the organism will receive either an *excite* or an *inhibit* internal stimulus (corresponding for instance to a reward or punishment) that will lead in turn to a reinforcement or a rejection of the action.

Let us consider a simple thought experiment where a pigeon is learning to discriminate between grains and pebbles corresponding to two possible vectors \mathbf{I} of external visual stimuli. The circuit given in Fig. 2, represents the interaction of four threads **sense(I)**, **learn(accept(I))**, **accept(I)** and **reject(I)**, together with two threads **ltp** and two opposite threads **ltd** (for long term depression). In addition to the external stimuli, this circuit incorporates two internal stimuli **excite(accept(I))** and **inhibit(accept(I))** that correspond to feedbacks from probing the food according to a predefined set of accepted elements.

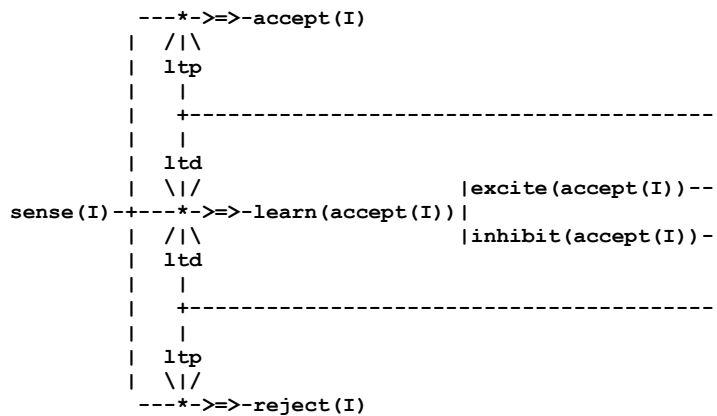


Fig. 2. A meso scale virtual circuit implementing simple operant conditioning

At the beginning of the simulation, the pathways from **sense(I)** to **learn(accept(I))** is open, while the pathways to both **accept(I)** and **reject(I)** are closed. After a few trials, the pigeon will have learned to close **learn(accept(I))** and to open either **accept(I)** or **reject(I)**. This process matches a fundamental principle in circuit neuroscience according to which *inhibition* in neuronal networks during baseline conditions allows in turn for *disinhibition*, which then stands as a key mechanism for circuit plasticity, learning, and memory retrieval [20].

2.1.3 Representing virtual circuits by symbolic expressions

The circuit given in Fig. 1 gives rise to the following symbolic expression:

```
threads (aplysia (reflex)) :
  [thread (sense (us) ,
    [fire (ltp (sense (cs) ,motor (cs)) ) ,
      send (motor (us) ) ] ) ,
    thread (sense (cs) ,
    [merge (ltp (sense (cs) ,motor (cs)) ) ,
      send (motor (cs) ) ] ) ,
    thread (motor (us) ,
    [receive (sense (us) ) ,
      effector (motor (us) ) ] ) ,
    thread (motor (cs) ,
    [receive (sense (cs) ) ,
      effector (motor (cs) ) ] ) ,
    thread (ltp (Q,R) ,
    [join (Q) ,
      increment (weight (Q,R) ) ] ] ] .
```

In this example, the instruction tree of each thread reduces to a sequence of virtual instructions such as **fire**, **send**, **merge**, etc, but more generally an instruction tree can contain *alternatives* commanded by *guards*. Formally, expressions for instruction trees belong to a language S whose BNF syntax is:

```
<tree> ::= [] || <sequence> || [<alternative>]
<sequence> ::= [<instruction>|<tree>]
<alternative> ::= <branch> || (<branch>;<alternative>)
<branch> ::= (<guard>|<tree>)
```

Whereas the non-terminal symbol $\langle\text{guard}\rangle$ represents conditions derived from internal stimuli, $\langle\text{instruction}\rangle$ stands for virtual machine instructions. This language S of instruction trees is not to be confused with the logical language L that will be used to define *virtual code implications* into which instruction trees will be then compiled.

2.1.4 Compiling instruction trees into virtual code implications

Virtual code implications are compiled from thread expressions and have the form

$$\text{Guard} \Rightarrow T:\text{Instruction}$$

where *Instruction* is a virtual machine instruction and T its clock time. As an example, virtual code implications compiled from the above thread **sense (us)** are

```
true => 1: fire (ltp (sense (cs) ,motor (cs)) )
true => 2: send (motor (us) )
true => 3: end
```

In this simple example, successive clock time values (i.e., 1,2,3) correspond to a linear list traversal with no guards. More generally, this will give rise to a descent into trees containing guards.

2.1.5 Microcircuits implementing synaptic plasticity

Virtual circuits rely on communication protocols that are pictured in thread diagrams by iconic symbols representing themselves microcircuits. These protocols can be

defined by means of procedures that operate in pairs:

- **send/receive** denoted by $\rightarrow\Rightarrow$ or $\Leftarrow\Leftarrow$
represents a synaptic transmission
 - **join/merge** denoted by $/|\backslash$ or $\backslash|/$
implements long term potentiation/depression (**ltp/ltd**)
 - **push/pull** denoted by $\rightarrow\langle A \rangle\leftarrow$
models a short term cache memory
 - **store/retrieve** denoted by $\rightarrow\{P\}\leftarrow$
models an associative memory based on long term storage and retrieval (**lts/ltr**) .
- These protocols are detailed in [15] together with their corresponding microcircuits.

2.2 Top down construction of a virtual machine

Let us consider a set of fibers together with sets of initial weights for pairs of communicating threads within fibers and sets of elements accepted by fibers, and let *Model* designate the state of the virtual machine associated with these sets. The machine itself functions as non deterministic learning automaton that operates on the constrained neural substrate represented by *Model*. Formally, it consists in repeating a *sense-react* cycle of embodied cognition defined by the following *run* procedure:

```
run(Model)
  loop sense(Model)
      react(Model)
```

At the next level below, the *sense* procedure reflects the triggering of spike trains directed to sensory neurons. After possibly capturing an interrupt from sensors directed to a given active fiber, or *stream*, it updates *Model* using a transition function *input*:

```
sense(Model)
  if interrupt(Stream(Input))
  then Model ← input(Model(Stream),Input)
```

The function *input* first terminates the interrupted stream by clearing all its registers and queues and then resets the clocks of the sensory threads associated with sensors. The *react* procedure consists of a loop calling on each active thread in any stream to first deduce a virtual machine instruction and then update *Model* using a transition function *output*:

```
react(Model)
  for each Stream(Thread), T:Instruction,
  such that ist(Model(Stream)(Thread), (clock(T), T:Instruction))
  do Model ← output(Model(Stream)(Thread), T:Instruction)
```

The transition function *output* corresponds to the execution of virtual machine instructions implementing communication protocols. The *ist* predicate (standing for “is true”) implements *contextual* deduction [16]. Clock register values *T* are used to deduce, for each active thread, the next instruction satisfying the guard. Whenever a transition initiated by a thread succeeds, the thread clock is advanced and the next

instruction is deduced and executed, and whenever it fails, the current instruction is executed again i.e., the transition is attempted until it eventually succeeds. Altogether, this amounts to descending into an instruction tree, with its local clock time corresponding to the currently reached depth. As postulated by Zeki 2015 [21], there is no central clock, leading thus to the modeling of the brain as a massively asynchronous, parallel organ . The complete specifications of this machine are given in open access in [15].

3. Results

Previous application results [15] include a simulation of successive models of animal awareness up to its third level according to Pepperberg & Lynn 2001 [22]. Briefly, the first level of animal awareness corresponds to the ability to follow a simple rule involving the perception of a specific item or event and then either its acceptance or its rejection (i.e., a case of *matching/oddity* to sample). Whereas this first level does not allow for an immediate transfer to a similar task, an organism with the second level is aware enough of a rule to transfer it across situations and thus to adopt for example a *win/stay lose/shift* rule. The third level of animal awareness provides an organism with the additional capacity to integrate two different sets of stored information in order to make a categorical judgment (e.g., to *sort* items). The results presented below lie outside of this hierarchy and propose evolutive simulations of analogical reasoning, on one hand, and of memory awareness, on the other.

3.1 Modeling evolutive cases of analogical reasoning

3.1.1 Learning a simple analogical inference schema

Let us first consider a simple analogical inference schema involving two predicates p and q applied to objects $x1$ and $x2$, i.e.

$\{p(x1)\}$		$\{big(dog)\}$
$\{p(x2)\}$	e.g.,	$\{big(bear)\}$
<u>$q(x1)$</u>		<u>$strong(dog)$</u>
$q(x2)$		$strong(bear)$

where $\{F\}$ represents a fact F , or proposition, that has been previously memorized.

This inference schema can be viewed as first inducing an implication i.e.,

$$p(x) \rightarrow q(x)$$

where x is a variable, and then applying modus ponens i.e.,

$$\begin{array}{l} p(x) \rightarrow q(x) \\ \underline{p(x)} \\ \hline q(x) \end{array}$$

The corresponding circuits (where A, B are parameters defining a context e.g., **left**, **right**, and I, J vectors of percepts representing properties p, q) are given in Fig.3 and 4. In Fig. 3, both circuits build an *associative long term memory* of perceived facts. More precisely, facts $p(x1)$ and $p(x2)$, as perceived by the threads

sense (**A** (**X1** (**I**))) and **sense** (**B** (**X2** (**I**))) are first learned through a simple case of operant conditioning (see section 2.1.2) and accepted by the threads **see** (**A** (**X1** (**I**))) and **see** (**B** (**X2** (**I**))), which are then memorized via a *long term storage* (**lts**) process into a memory trace **{see** (**A** (**X1** (**I**)))} and **{see** (**B** (**X2** (**I**)))}

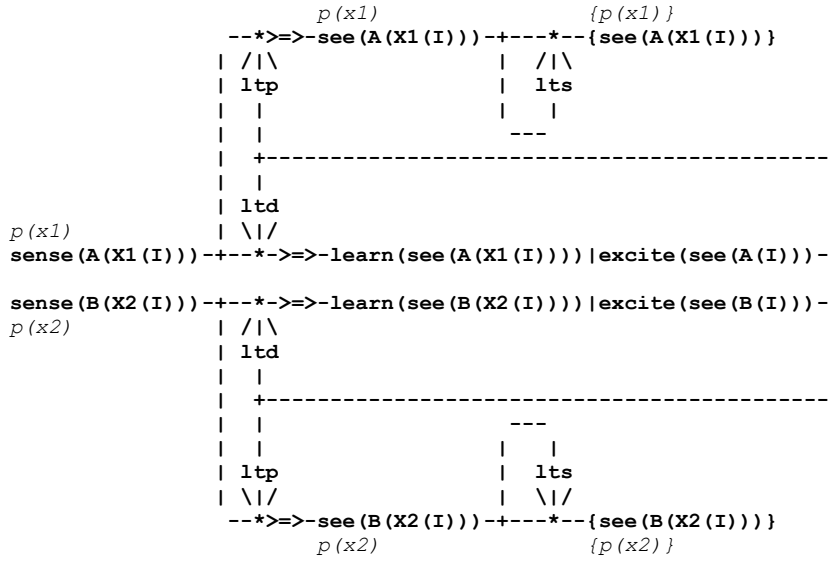


Fig. 3. Virtual circuit for memorizing perceptions

In Fig. 4, a circuit pattern representing an implication is first build in the upper half and then applied in the lower half to make an inference. More precisely, fact **q** (**x1**), as perceived by the thread **sense** (**A** (**X1** (**J**))), is first learned through a simple case of operant conditioning and accepted by thread **see** (**A** (**X1** (**J**))). In parallel, fact **p** (**x1**) is recalled from its trace **{see** (**A** (**X1** (**I**)))} via an **ltr** process and classically conditioned (see section 2.1.1) to infer **q** (**x1**) in thread **infer** (**A** (**X1** (**J**))). Simultaneously (or rather, slightly after), fact **q** (**x2**) is perceived by thread **sense** (**B** (**X2** (**_**))) carrying a non instantiated *property* and is not accepted. In parallel, fact **p** (**x2**) is recalled from its trace **{see** (**B** (**X2** (**I**)))} into thread **recall** (**B** (**X2** (**I**))), which is conditioned to be eventually matched with **p** (**x1**) and then to infer **q** (**x2**) in thread **infer** (**B** (**X2** (**J**))). Finally, thread **sense** (**B** (**X2** (**_**))) is conditioned to be accepted in thread **see** (**B** (**X2** (**J**))), thus representing a contextual and/or constrained reactive processes relating a perception to an action.

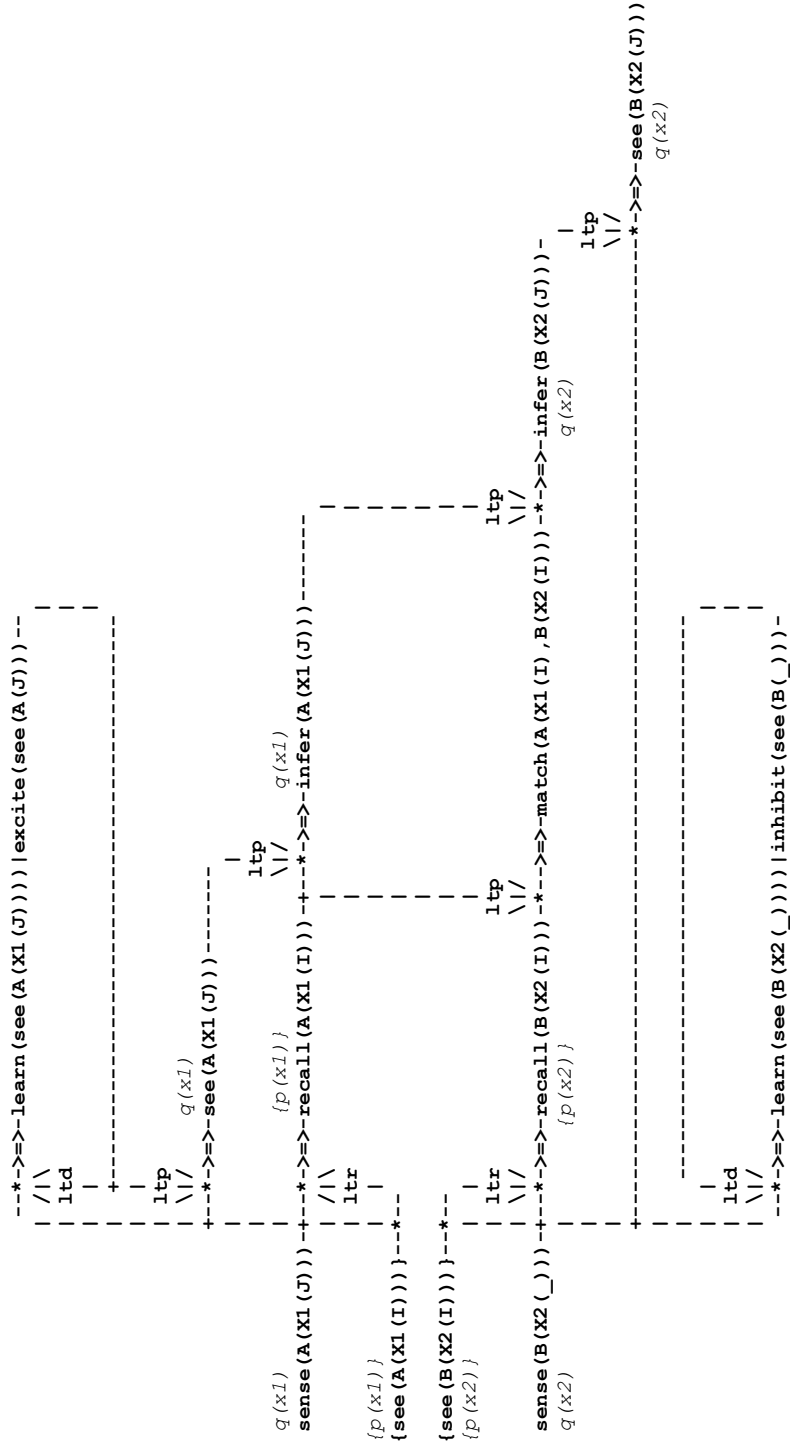


Fig. 4. Virtual circuit for implementing a simple case of analogical inference

3.1.2 Learning an analogical inference schema implementing transitive relations

Let us now consider a case of relational inference based on transitive relations i.e.,

{p(x1, y1), p(y1, z1)}	{smaller(mouse, cat), smaller(cat, dog)}
{q(x2, y2), q(y2, z2)} e.g.,	{higher(tree, house), higher(house, car)}
<u>p(x1, z1)</u>	<u>smaller(mouse, dog)</u>
q(x2, z2)	higher(tree, car)

This inference consists in first inducing a second order implication representing a generic transitive relation, where P, X, Y, Z are variables

$$P(X, Y), P(Y, Z) \rightarrow P(X, Z)$$

and then applying modus ponens

{P(X, Y), P(Y, Z) → P(X, Z)}
P(X, Y)
<u>P(Y, Z)</u>
P(X, Z)

Let us extend the definition of the perceptual context by adding two parameters C, D e.g., **front**, **rear**. The circuit implementing this inference schema is given in Fig. 5. As a distinctive difference from the previous circuit, where properties in remembered facts $\{p(x1)\}$ and $\{p(x2)\}$ need to match, the circuit in Fig. 5 relies on facts $\{p(x1, y1), p(y1, z1)\}$ and $\{q(x2, y2), q(x2, z2)\}$ with properties that don't need to match. Whereas the analogical inference implemented in Fig. 4 allows for the cognitive transfer of a *property*, the circuit of Fig. 5 allows for the transfer of an *object*. Consequently, the conditioning step leading to $\text{infer}(D(_K), Z2(L))$ in Fig. 5 still carries a non instantiated object that can be instantiated only in the final step relating $\text{sense}(D(X2(_), Z2(_)))$ to $\text{see}(D(X2(K), Z2(L)))$.

3.1.3 Combining analogical inference schemas

Let us finally turn to an example of relational inference of the following type

{p(x1, y1), q(y1, z1)}	{father(bill, mary), mother(mary, sam)}
{p(x2, y2), q(y2, z2)} e.g.	{father(tom, cathy), mother(cathy, jack)}
<u>r(x1, z1)</u>	<u>grandfather(bill, sam)</u>
r(x2, z2)	grandfather(tom, jack)

This extension involves the two types of cognitive transfers just considered. A combination of the two corresponding circuits is however far from being straightforward, the difficulty being here the parallel matching of multiple interleaved properties. Whereas behaviors relying on simple analogical reasoning and transitive inference, as modeled by the circuits of Fig. 4 and 5, have been observed in animals, this more complex example is unarguably out of their reach. It is interesting to note that previous modeling approaches relying on substitutions [8] have led to similar conclusions [23].

3.2 Modeling a simple case of meta-cognition

The additional simulations presented below pertain to the domain of *meta-cognition*. In broad terms, meta-cognition refers to "monitoring and control processes that allow for maintaining a self-referential check and balance" (Fleming & al. 2012 [24]). A simple case of meta-cognition is constituted by *memory awareness*. When applied to humans, this allows them in particular to feel certain or uncertain, and ultimately, "to know when they do not know" (Smith & al. 2012 [25]). Some non-human animals (thereafter, animals), who selectively decline tests when their memory is weak, seem to exhibit such capabilities. In order to validate this hypothesis, Templer and Hampton 2012 [26] designed a two phase experiments. In the first phase, monkeys are trained to take a "memory test". Each trial in this phase consists of an *information* stage, in which a randomly chosen cup is openly baited in front of the subject and then momentarily removed from its view, followed by a *choice* stage, in which the subject is asked to retrieve the food. In the second phase of this experiment, a *decline* option (also commonly referred to as the "uncertainty response") is introduced. This option consists in a fixed cup offering a lesser, but guaranteed reward.

Actually, much on the debate around meta-cognition does focus on the following question: can animals' uncertainty responses be interpreted within an associative framework, or should they be considered as the expression of a higher-level cognitive mechanism? An assessment of the research in this field has been proposed by Dickinson 2012 [27], who argues that mechanisms based on constrained associative processes can implement psychological rationality. By pointing out to an issue originally raised by Holland 1990 [28], he raised the possibility of using *retrospective revaluation* (Dickinson & Burke 1996 [29]): whereas simple associative theory assumes that learning occurs when a stimulus is present, retrospective revaluation allows for learning about associatively retrieved representations (see below in section 3.2.2).

3.2.1 Modeling a reliable memory.

Let us consider Templer and Hampton's memory test [26] and let us first assume a perfectly reliable memory without a decline option. Without any loss of generality, two possible cups only i.e., **A** and **B**, are taken into account here. This memory test can be implemented (see the circuit in Fig. 6) as an extended case of *operant conditioning* with a choice between two alternatives **A** and **B** captured by the random internal stimuli **fetch(A)** and **fetch(B)** and coupled with a recall from a short term memory **<A>** and ****. It is interesting to note that this circuit matches the one used in [15] to implement the second level of animal awareness: in both cases, subjects are trained to solve a task by learning a *win/stay lose/shift* rule.

3.2.2 Modeling an unreliable memory

According to Templer and Hampton 2012 [26], memory awareness provides subjects with the capacity to make "judgments" about the quality of their memory e.g., to discriminate between strong and weak memories. Let us further quote Templer and Hampton literal description of how monkeys possibly use their memory awareness when confronted with a decline option: "*When memory is weak, monkeys with memory awareness should opt for the decline response rather than running the risk of receiving no raisins by guessing where the two raisins were located. When memory is strong, and they have a high probability of choosing the correct choice cup, they should select the remembered cup*". This description, which distinguishes the correct cup (i.e., the one that actually holds the preferred reward) from the remembered cup (i.e., the one the memory has, correctly or incorrectly, associated with the preferred reward), can be rewritten under the form of two rules:

if memory is weak

- > *there is a low chance of choosing the correct cup*
- > *opt for the decline response*

if memory is strong

- > *there is a good chance of choosing the correct cup*
- > *select the remembered cup.*

These two rules rely on the definition and access to a mental state, representing in this case a memory that besides holding an objective content (i.e., the remembered cup) is subjectively perceived as either *weak* or *strong*. Obviously, this perception involves a wide range of cumulative phenomena (such as distraction, fatigue, or other decay mechanisms), and thus cannot be reduced to a single deterministic value e.g., "*weak*" or "*strong*". A possible way to represent and apply these rules is to use a probability distribution as follows:

in order to represent a weak memory

- > *assign a low probability to the correct location*
and complementary higher probabilities to the other options
- > *perform a weighted random choice.*

A strong memory can be similarly represented by assigning a high probability to choosing the correct location. Retrieving from the memory then reduces to weighted random choices. For this scheme to be consistent, the "remembered location" is an elusive concept that needs to be replaced by the association of the correct location with its probability. In other words, the physiological mechanism that creates the short-term memory (i.e., **-<A>-** in our wiring model) is supposed to be reliable, but the process of retrieving from this memory is degraded by cumulative factors resulting in an unreliable *noisy transmission*.

Model of an unreliable memory without a decline option

Let us now consider a model of unreliable memory, but without yet a decline option, as given in Fig. 7, where additional wiring is in italics. In neural terms, two interneurons **noise (A| [A,B])** and **noise (B| [A,B])** open a new pathway between the memory neurons **recall (A| [A,B])** and **recall (B| [A,B])**, on one hand, and two new transmitter neurons **filter (A| [A,B])** and **filter (B| [A,B])**, on the other. Furthermore, whereas the internal stimuli **fetch (A)** and **fetch (B)** attached to **learn** are purely random, the stimuli **catch (A)** and **catch (B)** attached to **filter** to reflect a noisy transmission integrate multiple physiological sources and, as argued above, follow from weighted random choices.

Model of an unreliable memory with a decline option

Memorizing a decline option at location **C** constitutes another example of an *associative long term memory* given by a memory trace (see Fig. 3 of section 3.1.1), in this case **{accept (C)}**. Introducing this option as an additional choice gives rise to a new layer of noisy transmission. This scheme (see Fig. 8), which allows for animals to learn about directly perceived stimuli as well as retrieved representations such as the decline option, constitutes a particular case of *retrospective reevaluation* [29] alluded to above. Turning out to look like an extension of the second and third levels of animal awareness combined together, it shows how a simple case of meta-cognition can be reduced to successive layers of associative memories.

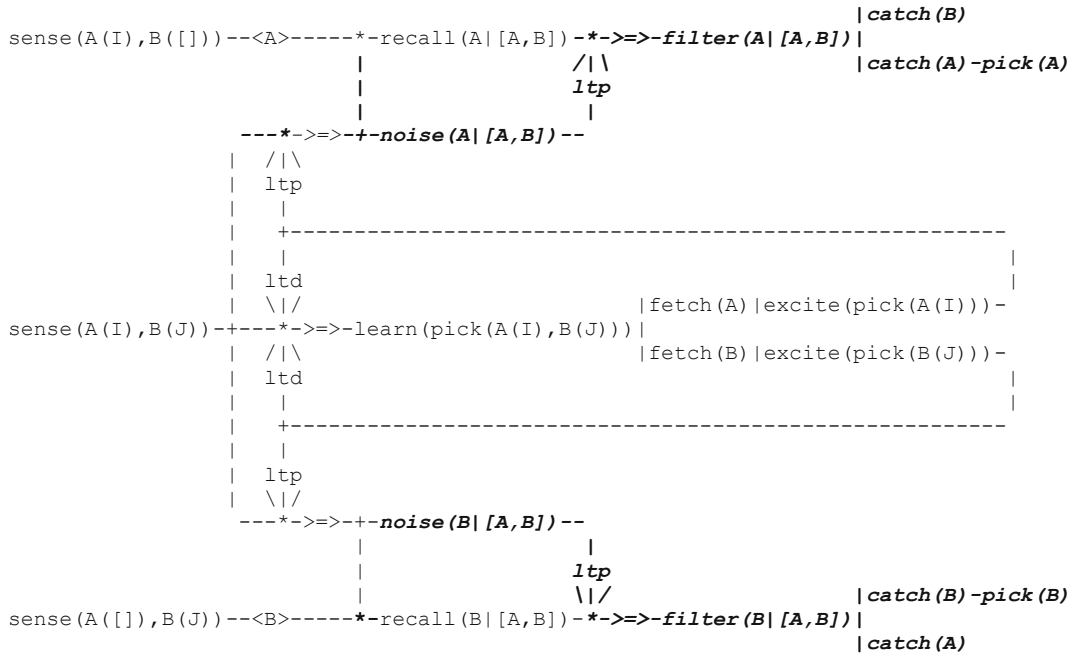


Fig. 7. Circuit implementing an unreliable memory without a decline option

4 Discussion and conclusion

Various proposals have been made to close the gap between the level of individual neurons and higher levels supporting behavior. A possible solution is to consider group of neurons, or *neural assemblies*. It is proposed here to model neural assemblies in a simulation framework driven by a virtual machine linking neural dynamics to behaviors. While the usual approach to simulating neural dynamics starts with current flows represented by differential equations, we opted for a conceptual abstraction of synaptic plasticity represented by communicating processes. As a consequence, there is no reference to any specific neural network model. While these simulated neural structures are general-purpose entities, symbolic models of cognition constrain them into specific-purpose processes driven by a virtual machine. As reflected by the terms in the title of [15] (“i.e., linking *neural dynamics to behaviors* through *asynchronous communications*”) and of this contribution (linking *perception and action* through *symbolic models of cognition*), this virtual machine clearly stands as an interface between the neural and cognitive levels.

Following previous results on modeling animal cognition up to the third level of animal awareness, this formalism has been applied in section 3.1 to simulate the learning of evolutive analogical inference schemas involving two kinds of cognitive transfer, namely that of an object and that of a property. Whereas schemas involving one kind of transfer correspond to cognitive abilities observed in animals, schemas involving a combination of these two kinds correspond to abilities presumably enjoyed by humans only. As these circuits show, higher level cognitive facilities can emerge through learning schemas involving in turns both classical and operant conditioning. If indeed classical and operant conditioning may be combined into hybrid models, then (as speculated by Carew2002 [30]) “*an exciting principle might emerge: evolution may have come up with a neural 'associative cassette' that can be used in either type of conditioning, depending of the neural circuit in which it is embedded*”. Altogether, this approach to brain modeling could thus lead to an actualization of the old concept of behaviorism [2], which was founded through the experimental exploration of these two types of conditioning.

The same formalism has been applied in section 3.2 to simulate memory awareness as a case of retrospective reevaluation. This result shows that simple cases of meta-cognition can be implemented by successive layers of associative processes reflecting noisy transmissions.

In conclusion, it is argued that these results could lead to a reconsideration of the whole concept of a “neural code” relating perception and behavior. Such a neural code may well reside in the spatial arrangement of mesoscale circuit patterns (i.e., a kind of population or sparse coding, as opposed to the more traditional rate or temporal coding associated with spike trains). One might then even consider that there is actually no code at all (in the sense of a specific arrangement always associating the same response to a given stimulus), and that “the code is the overall structure itself”. More precisely, perception might be related to behaviors through the paths found by evolution via iterative hebbian learning.

References

1. Miller, G. (2003). The cognitive revolution: a historical perspective. *Trends in Cognitive Sciences*. 7 (3).
2. Skinner, B (1953). *Science and Human Behavior*". MacMillan
3. Thagard, P, (2005). *Mind: Introduction to Cognitive Science*, MIT Press.
4. Anderson, J R et al (2004). An integrated theory of the mind. *Psychological Review*, 111(4), 1036–1060.
5. McCulloch, W, Pitts W (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 7, 115-133.
6. Hodgkin, AL, Huxley, AF (1952). A quantitative description of membrane current & its application to conduction & excitation in nerve. *Journal of Physiology*, 17(4), 500-544.
7. Shastri, L, Ajjanagadde, V (1993). From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, 16, 417–51W.
8. Hummel, J, Holyoak, K. (2003). A Symbolic-Connectionist theory of Relational Inference and Generalization. *Psychological Review* 110 (2), 220-264.
9. Jilk, D, Lebiere, C, O'Reilly, R, Anderson, JR (2008). SAL: An explicitly pluralistic cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 20(3). 197-218.
10. Vernon, D (2014). *Artificial cognitive systems: a primer*. The MIT Press.
11. Brooks, R (1991). Intelligence without representation. *Artificial Intelligence*, 47, 139–159.
12. van der Velde , F, de Kamps, M (2015). The necessity of connection structures in neural models of variable binding. *Cognitive Neurodynamics* 9, 359–37.
13. Marr, D (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Freeman.
14. Frégnac, Y, Bathellier, B (2015). Cortical Correlates of Low-Level Perception: From Neural Circuits to Percepts. *Neuron* 88.
15. Bonzon, P, (2017). Towards neuro-inspired symbolic models of cognition: linking neural dynamics to behaviors through asynchronous communications. *Cognitive Neurodynamics* doi:10.1007/s11571-017-9435-3.
16. Bonzon, P, Cavalcanti, M, Nossum, R. (eds). (2000). *Formal Aspects of Context*, Applied Logic Series 20 , Kluwer Academic Publ.
17. Palm, G (1982). *Neural Assemblies. An Alternative Approach to Artificial Intelligence*. Springer.
18. Kandel, E, Tauc, L (1965). Heterosynaptic facilitation in neurones of the abdominal ganglion of *Aplysia depilans*. *Journal of Physiology (London)* 181.
19. Hebb, D (1949). *The organization of behavior. A neuropsychological theory*. J. Wiley.
20. Letzkus, J, Wolff, S, Lüthi, A (2015). Disinhibition, a Circuit Mechanism for Associative Learning & Memory. *Neuron*, 88(3), 264–276.
21. Zeki, S, (2015). A massively asynchronous, parallel brain, *Phil. Tran. R. Soc. B*, 370:20140174.
22. Pepperberg, I, Lynn, S (2000). Possible Levels of Animal Consciousness with Reference to Grey Parrots (*Psittacus erithacus*). *American Zoologist*, 40, 893-901.
23. Penn, D, Holyoak, K, Povinelli, D (2008). Darwin's mistake: Explaining the discontinuity between human and nonhuman minds. *Behavioral and Brain Sciences*, 31, 109–178.
24. Fleming, S, Dolan, R, Frith, C (2012). Metacognition: computation, biology and function. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367 1280-1286.
25. Smith, J, Couchman, J, Beran, M (2012). The highs and lows of theoretical interpretation in animal-metacognition research. *Philos Trans R Soc Lond B Biol Sci*. 367(1594): 1297–1309.

26. Templer, V, Hampton R (2012). Rhesus monkeys (*Macaca mulatta*) show robust evidence for memory awareness across multiple generalization tests. *Animal Cognition* 15(3), 409-419.
27. Dickinson, A (2012). Associative learning and animal cognition. *Philos Trans R Soc Lond B Biol Sci.* 367(1603), 2733–2742.
28. Holland, P (1990). Event representations in Pavlovian conditioning: image and action. *Cognition* 37, 105–131.
29. Dickinson, A, Burke, J (1996). Within-compound associations mediate retrospective revaluation of causality judgements. *Quart J of Exp Psychology*, 49B, 60-80.
30. Carew, TJ (2002). Neurology, Understanding the consequences. *Nature New & Views*, 407, 803-806.