# Cognition, Concurrency Theory and Reverberations in the Brain: in Search of a Calculus of Communicating (Recurrent) Neural Systems

**H. Bowman* and Li Su****

***Centre for Cognitive Neuroscience and Cognitive Systems, School of Computing, University of Kent, Canterbury, Kent, UK**

****Experimental Psychology, University of Cambridge, Cambridge, UK**

**Abstract**

**We consider whether techniques from concurrency theory can be applied in the area of Cognitive Neuroscience. We focus on two potential applications. The first of these explores structural decomposition, which is effectively assumed by the localisation of function metaphor that so dominates current Cognitive Neuroscience. We take concurrency theory methods, especially Process Calculi, as canonical illustrations of system description notations that support structural decomposition and, in particular, encapsulation of behaviour. We argue that carrying these behavioural and notational properties over to the Cognitive Neuroscience setting is difficult, since neural networks (the modelling method of choice) are not naturally encapsulable. Our second application presents work on verifying stability properties of neural network learning algorithms using model checking. We thereby present evidence that a particular learning algorithm, the Generalised Recirculation algorithm, exhibits an especially severe form of instability, whereby it forgets what it has learnt, while continuing to be trained on the same pattern set.**

## Introduction

**Historically, cognitive psychology has been significantly influenced by computer science in general and theoretical computer science in particular. Obvious examples of this interplay would include the following. (1) Computational logic has occupied a central position in symbolic theories of thought, e.g. Johnson – Laird's Mental Models [1], Fodor's criticism of connectionism [2] or, more broadly, production system architectures of mind [3]; this is**

**the Good Old Fashioned Artificial Intelligence (GOFAI) tradition. (2) In consideration of cognition, recourse has often been made to computability and limits thereof, e.g. Penrose's argument for the incomputablity of consciousness [4], the perspective that brains are uninteresting – just "arbitrary" Turing complete computational devices [2, 5] and "functionalist" perspectives on consciousness [5]. (3) Indeed, in original conception, the term cognitive was explicitly coined (in opposition to behaviourism) with reference to (symbolic) computational explanations of thought [6].**

**Neuroscience has also traditionally been influenced by computer science, but more in respect of subsymbolic and statistical learning techniques, e.g. the linking of Hebbian learning to principal component analysis [7] or neural coding schemes to Kohonen maps [8].**

**It might then seem that more recent computer science methods, arising in the domain of formal methods and concurrency theory, e.g. [9-11], could also make a contribution to understanding mind and brain. Indeed, the brain is clearly a massively parallel system in which (despite proposed roles of the thalamus as a global synchroniser) control, at a physical substrate level, would seem to be broadly distributed. In addition, many theories of cognition are explicitly parallel and distributed, e.g. Barnard's Interacting Cognitive Subsystems [12] and Baddeley and Hitch's working memory model, where for instance, the Phonological loop and Visual Spatial Sketchpad evolve in simultaneous independence [13]. In particular, the pervasiveness of parallelism and distribution suggests that the techniques developed under the concurrency theory banner should be of great relevance to the current cognitive neuroscience project, which explicitly seeks to relate mind to brain [14]. This is the issue we consider in this paper.**

**A point to note though, is that the requirement to explicitly consider how the brain might realise particular cognitive functions, suggests a level of neurophysiological realism beyond that offered by the most basic traditional connectionist models, e.g. [15]. One such aspect of realism that has major consequences for computational complexity is *recurrence*. That is, the brain is densely recurrently connected. For instance, in addition to major "bottom-up" processing pathways, i.e. *from* sensory systems, there are also many "top-down" projections, from hierarchically higher areas *back down* towards sensory regions. Although there are trade-offs in all abstraction choices, modelling that abstracts**

**away this mechanistic complexity risks identifying models that could not plausibly obtain in a real brain.**

**The practical consequences of faithfully incorporating recurrence in neural models are though severe. The dynamics of bidirectional neural networks are massively more complex and difficult to characterise than unidirectional networks. In particular, the combination of nonlinear activation dynamics and bidirectionality makes confirming the stability of nontrivial recurrent networks a serious analytical undertaking, e.g. see [16] for a verification of stability of a model of V1 (i.e. visual area one) containing recurrence.**

**In this setting, we consider two particular ways in which concurrency theory might contribute to the cognitive neuroscience project, with others highlighted in the discussion. These two areas are as follows.**

1) **Structural decomposition in architectural theories of mind and brain.**
2) **Verification of stability in recurrent neural models.**

**We consider these in turn.**

## Structural Decomposition

### Decomposition in Concurrency Theory

**One of the foundational observations of concurrency theory, and particularly process calculi, was of the noncompositionality of first generation formalisms for specifying concurrent and distributed systems. For instance, Milner criticised Petri Nets in this respect [17]. The key issue being that Petri Nets are structurally flat: one cannot explore them isolated-component by isolated-component and accordingly one cannot, at least not directly, construct systems encapsulated-component by encapsulated-component. Thus, when a new component is added to an existing Petri Net, the entire Net needs to be redesigned. The response inherent in process calculi, was to introduce a parallel composition operator, which allowed arbitrarily complex components with their own distinct "thread of control" to explicitly execute in parallel [10, 18, 19]. This was the origin of the notion of a process.**

**Indeed, concern for the challenges of system and software engineering, especially those associated with the problem of scale, were to the fore in the design of process calculi. That is, the basic operators and semantic underpinnings of these notations were explicitly devised to support the specification and design of *large* parallel and distributed systems. Particular examples of this consideration include definition of refinement relations [9, 10, 20] and more recent work on viewpoints-based system development [21, 22]. Most significantly, though, there was a specific consideration for the incremental (stepwise) description and, ultimately, construction of such systems. This can be seen in the principal operators of a process calculus such as LOTOS [10, 23], with similar constructs being present in CSP [9], CCS [18] and ACP [24]. For example, in the manner previously suggested, the LOTOS parallel operator composes separately threaded processes, subject of course, to synchronous message passing communication; hiding conceals the internal details of a process's implementation, isolating that implementation and enabling specifiers to preserve a process's behaviour when placed in a context; and interaction between components is explicitly controlled through gate lists, which effectively prescribe interfaces of allowable communication. As an illustration, the top-level specification of the protocol in figure PROTOCOL, would directly reflect its component structure, as follows,**

***hide receiveAck, send, receive, sendAck in***
***(Sender [get, receiveAck, send]***
***III***
***Receiver [put, receive, sendAck])***
***|[receiveAck, send, receive, sendAck]|***
***DupMedium [receiveAck, send, receive, sendAck]***

**where, *III* denotes independent parallel composition, i.e. with no interaction between composed processes and *|[G]|* denotes parallel composition with synchronous interaction on all gates in *G*.**

**As a result of these features of Process Calculi, (process) components can be *encapsulated*, preserving their behavioural essence across contexts. For example, a LOTOS specification of a stop-and-wait protocol remains a stop-and-wait protocol whatever**

**context it is composed with. True, there are contexts in which the protocol might be placed that would yield degenerate behaviour, e.g. deadlock or livelock, but that emergent behavioural outcome does not imply an alteration to the protocol's underlying behaviour; interaction *with* the protocol is in error, but the underlying behaviour of the protocol is unchanged.**

**Another aspect of process calculi that is so natural we almost do not notice its presence, is the possibility to nest components to arbitrary depth, i.e. not just decomposition into components at one level, but true (depth-unconstrained) *hierarchical* decomposition. For instance, the decompositional structure inherent in a communication protocol, such as that in figure PROTOCOL, would be directly reflected in the structure of a LOTOS specification of such a protocol, e.g. c.f. [10]. For instance, component *DupMedium* would contain two processes: *Medium* and *AckMedium*.**

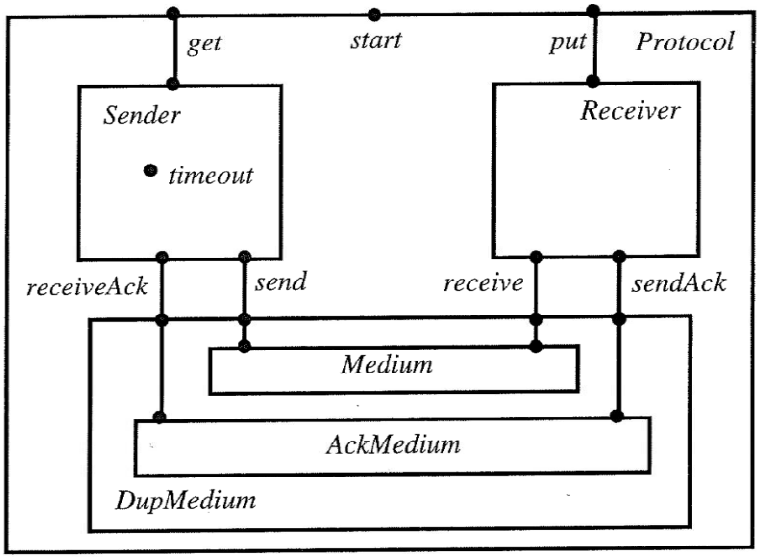


**Figure PROTOCOL: A communication protocol.**

**Clearly, such a capacity to keep on nesting and decomposing to whatever level is required, without reaching an artificial decompositional bottom-out, is critical when constructing large scale systems. That is, there is flexibility in system construction, with decompositional "brick-walls" avoided.**

**In addition to a calculus' operators, much theory development and debate has concerned what could broadly be called the compositionality properties of process calculi.**

**For example, equivalences and preorders have been sought that are congruences (respectively precongruences) over a calculus' operators [10, 18]. This ensures that a semantic relationship between two processes (e.g. their equivalence) is preserved when they are placed in the same context. For instance, for any context *C [.]*, an equivalence $\sim$ is a congruence if when $P \sim P'$, $C[P] \sim C[P']$. Incremental system development is particularly feasible in the presence of congruence relations, since relationships holding at the level of component parts are preserved when substituted into a composite whole.**

**Our basic point then, is that techniques available in concurrency theory, and in fact throughout computer science, are designed with the problem of scale in mind, i.e. with constructs and characteristics that enable large-scale systems to be described and analysed. The issue being that to enable big systems to be thought about and designed, there is a need to decompose into bite-size parts that can be considered in isolation, subject to controlled interaction through clearly defined interfaces. Such decomposition enables incremental, component-by-component, system development, providing what might be viewed as an engineering reductionism.**

**A key final point to note is that the system decomposition enabled in concurrency theory is not just structural and syntactic, it is also behavioural. That is, when a box is conceptually placed around a component, the *behaviour* within that component is encapsulated; in other words, the basic algorithm the component implements is secured and cannot be altered by instantiating it in a context. Such behaviour-encapsulating structural decomposition is critical to enabling incremental (stepwise) design of systems.**

## Decomposition in Cognitive Architectures

**A number of architectural theories of cognition sit very nicely with the compositionality perspective of concurrency theory in general and process calculi in particular. One such architecture is Barnard's Interacting Cognitive Subsystems (ICS) [12], see figure ICS.**

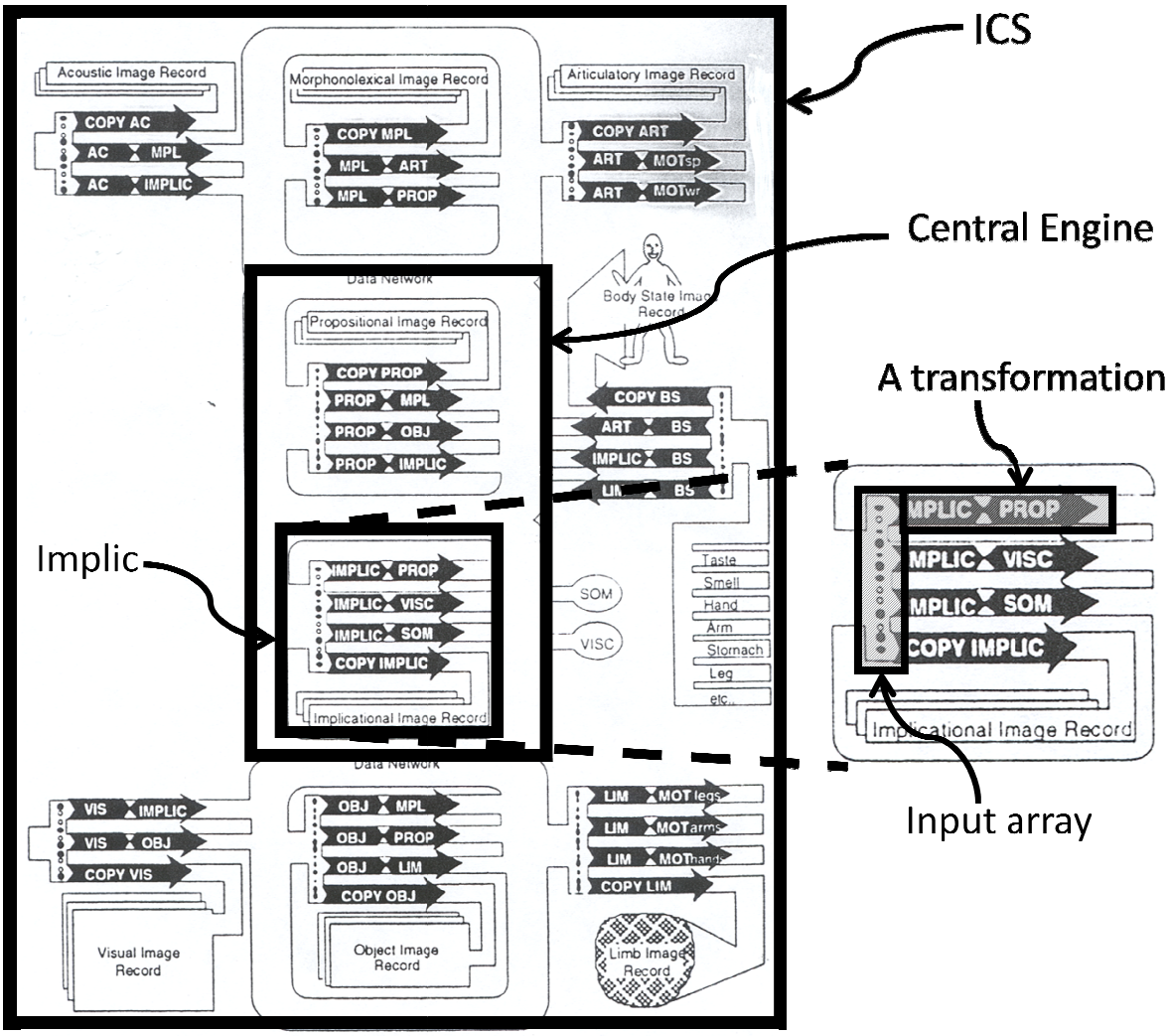


**Figure ICS: Structural decomposition in Interacting Cognitive Subsystems (ICS). One hierarchical decomposition path is illustrated by boxes with bold surrounding lines.**

**For a full (informal) description of the ICS architecture, see [12], but the point for us here is that there is clear structural decomposition: the Central Engine component sits within ICS itself, the implicational (Implic) subsystem sits within the Central Engine, and input array and transformation components sit within Implic, etc. In addition, subsystems evolve in parallel and none of them has a global view of the system state. Thus, control is at its core distributed. In addition, the notion of subsystem found in ICS sits quite naturally with behavioural encapsulation. That is, interaction between subsystems is through a restricted set of operations, with internal implementation details hidden and inaccessible from outside a subsystem.**

**Indeed, these aspects of ICS have made it a natural candidate for process calculus modelling. Accordingly, portions of the architecture have been formalised in LOTOS and analysed deductively and through simulation [25-27]. In addition, although not currently complete, there is no fundamental reason why the entire ICS architecture could not be formalised with this approach. In particular, an incremental (stepwise) construction using structural decomposition, in the manner so familiar to computer scientists, is quite feasible.**

**Although ICS is particularly suitable for concurrency theory formalisation, all models falling within the symbolic architecture of mind approach (e.g. SOAR [3], ACT-R [28] and EPIC [29]) should be similarly decomposable. Indeed, the symbolic nature of all cognitive modelling within the Good Old fashioned Artificial Intelligence (GOFAI) tradition would be expected to make structural decomposition feasible, whether the unit of decomposition be operations, functions, modules, objects or processes. For example, an architecture such as SOAR [3] contains a number of effectively encapsulated and independently evolving parallel components. Figure SOAR illustrates this decomposition; i.e. long-term memory, working memory, perceptual systems and motor systems are distinct, state independent, components.**

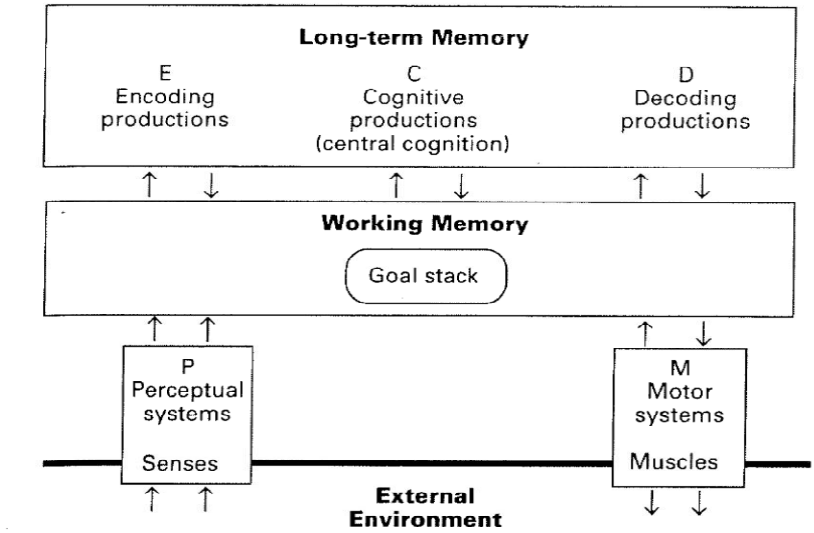


**Figure SOAR: Top-level structure of the SOAR cognitive architecture, as presented in figure 4.15 of [3].**

## Decomposition in the Brain

**In physical makeup, the brain also exhibits decompositional structure, with hierarchical nesting. This is apparent from coarse structural divisions (e.g. into two hemispheres) and cytoarchitecture (c.f. Brodman areas) [14]. The latter subdivision reflects physiological segregation in the brain; that is, discontinuities in the presentation of tissue identified with in vitro staining procedures; these might, for example, correspond to transitions in the density or type of constituent neurons. By way of illustration, figure BRAIN depicts the coarse decomposition arising from subdivision of the cortex by lobe and the finer partitioning arising from Brodman areas.**

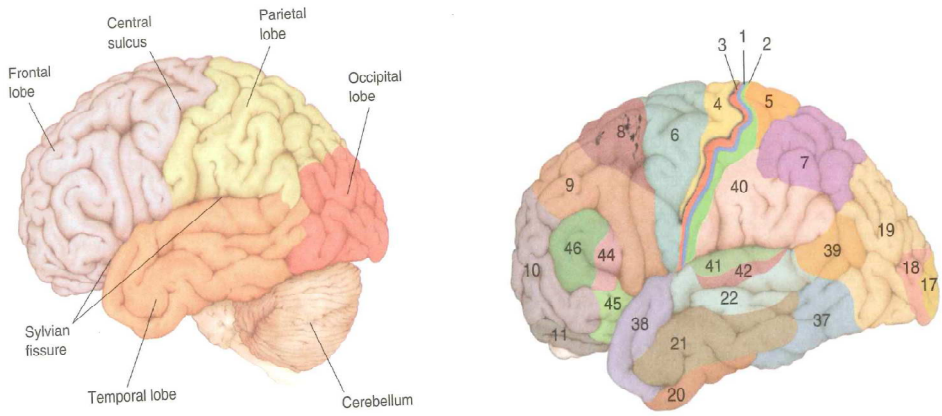


**Figure BRAIN: Left panel: The brain's lobes. Right panel: Brodman areas.**

**There are also evident finer divisions, such as widespread layering, e.g. the regular six laminar layout of the cortex [8], or the functional partitioning associated with cortical column [30], whereby, the neurons in a column orthogonal to the cortical surface, all exhibit similar tuning, e.g. all respond to the same region of space.**

**Furthermore, the central metaphor of current imaging neuroscience is exactly localisation of function, i.e. association of functional characteristics to brain regions, e.g. the hippocampus with formation of long-term memories [8] and the Amygdalar with detection of emotional salience, perhaps particularly threat [31]. Implicit to this activity is a belief that the brain's structural decomposition in physical form, can be married to a rich**

**decomposition of function, i.e. of behaviour, indeed, what might be described as computation. Effectively (in the metaphor's strongest form), a physical reductionism in one-to-one relationship with a functional reductionism is posited, see figure MAPPING for an illustration of this mapping ambition.**

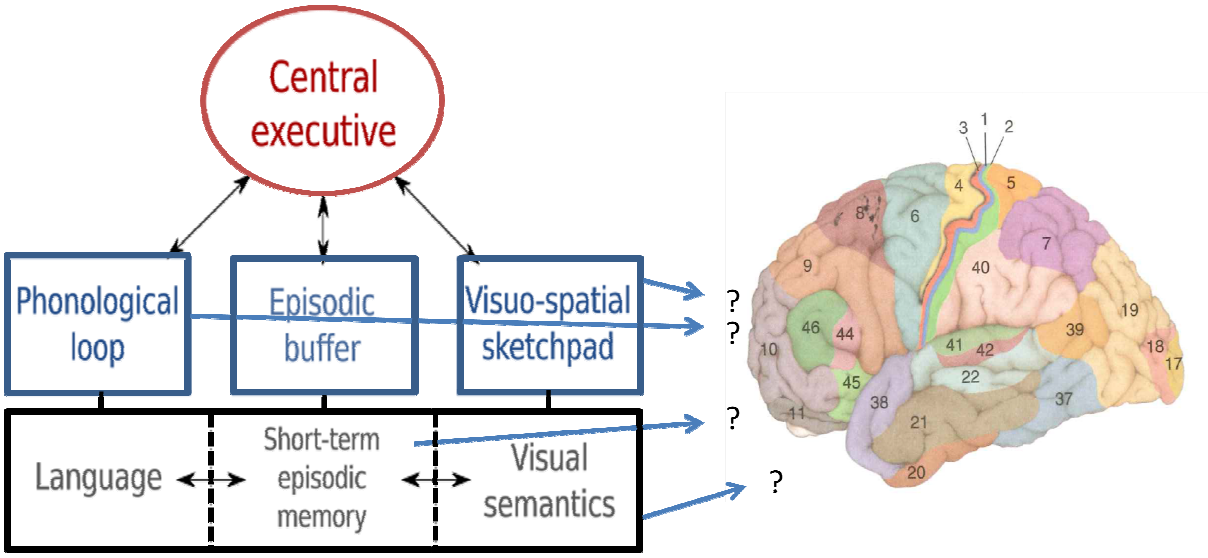


**Figure MAPPING: Mapping cognitive theories to brain regions. On the left here is one of the most successful theories in cognitive psychology, Baddeley and Hitch's working memory model [13]. A holy grail of cognitive neuroscience would be to associate components of such a cognitive theory to, say, Brodman areas.**

**Identifying neurophysiological correlates of informally defined cognitive theories (i.e. those that are effectively defined by diagrams and textual description] is certainly notable. However, associations between the structural decomposition in (formal) computationally-realised cognitive architectures and decomposition in the brain could, under the localisation of function hypothesis, obtain, and would be extremely compelling if convincingly identified; c.f. [32] for an attempt to make such an identification. Broadly speaking, it has, though, proved difficult to empirically verify such associations with current imaging techniques. Indeed, functional localisation claims, are no more than that and, apart from some well accepted localisations, particularly those focused on sensory and motor functions (the brain's peripheral systems, if you like), there remains considerable debate concerning most localisation hypotheses. Nonetheless, in terms of**

**our main interest here, functional localisation certainly does represent a very strong structural decomposition claim, which is at the heart of the Cognitive Neuroscience project.**

**Decomposition and Neural Behaviour**

***State of the Modelling Art*. Neural networks have become the modelling method of choice in the cognitive and brain sciences, especially when some link to neurophysioloigcal implementation is being claimed. However, it is important to note that the models developed with this method tend to be restricted in nature. Specifically, neural models in cognitive neuroscience seem almost exclusively to be small and very specialised. Admittedly, there is a class of neural models with large numbers of units: hundreds, thousands, or even, tens of thousands; see, for example, the classic "parallel distributed processing" tradition [15]. However, this scale tends not to be in respect of components (in the sense being considered here). Such models might, for example, contain thousands of nodes, but just 3 layers. It is these layers that are the components in the sense being considered here, with each having a distinct conceptual role in the understanding and construction of the system. Thus, the field seems swamped with models targeted at a restricted set of cognitive phenomena in a restricted psychological domain and containing a few layers/ components. For example, Machens et al. [33] presented a neural network model of the functioning of the frontal lobe, which may well be the most computationally, cognitively and neurophysiologically complex part of the brain. Indeed, the lobe does hold a significant proportion of the neurons in the brain. In addition, the paper is published in one of the most prestigious journals, Science. To make our position clear, we absolutely acknowledge the elegance and important contribution of the model presented. Our reason for highlighting the Machens et al. model is to illustrate the state of the art; that is, what is currently possible with available tools. Thus, our criticism is in no sense with the model itself. In this spirit, it has to be noted that the core model has just two units and simulates just one experimental finding (a short-term memory experiment on monkeys). Such small scale, highly targeted modelling can make major contributions to progressing understanding of focused scientific questions and is also, in this context, to be preferred under Ockham's razor; indeed we have published many such models ourselves, e.g. [34,**

**35]. However, in respect of the broad architecture of mind and its realisation in the brain, the contribution is more limited. In particular, Ockham's razor allows one to adjudicate between models with *similar* explanatory power. Our point is that existing neural models do not really provide an architectural-level explanation at all; in that context, Ockham's razor is not a relevant criterion to apply in the first place. This is the state of the art.**

**Clearly, it is the same mind that allows us to remember, to plan, to appreciated music and to comprehend and produce language, amongst many other capacities. However, any sense that these diverse faculties jointly constrain mind and brain is to a significant extent ignored by current neural network modelling in cognitive neuroscience.**

***The Curse of Decompositional Flatness*. In this context, it is interesting to consider how large architecture-level neural networks could be constructed. One might think that a calculus of communicating neural systems could be developed, in which neural network components would be composed together in the fashion that processes are in process calculi, see, for example, Milner's Calculus of Communicating Systems [18]. Indeed, one certainly could define a syntactic operator to wrap a neural network up as a component and further operators to wire such neural components together in parallel or series. However, this would just be syntax and, as motivated by structural decomposition in concurrency theory, to build systems or models incrementally, one has to be able to consider *behaviour* component-by-component. That though, requires certain, if you like, semantic properties to hold when composing components, the most fundamental of which is encapsulation of behaviour: when a stop-and-wait protocol is placed in a context, it stays a stop-and-wait protocol. This we argue, does not naturally arise in neural networks, and perhaps particularly in the variety of neural networks that satisfy the neurophysiological realism constraint so central to cognitive neuroscience.**

**As an illustration, consider the network in figure NET1 left panel. (Activation equations and parameter settings are presented in the appendix.) This is a simple Winner-Take-All (WTA) network; that is, the most strongly excited of the two units will, over time, win the competition, as reflected by its high activation and low activation of its competitor. Accordingly, the unit that will win will have the highest sustained net input, i.e. sum across its three weighted inputs, two inputs of which are extrinsic (i.e. from outside WTA) and the other intrinsic. Excitatory inputs will push a unit's net input up,**

**while inhibitory inputs will push it down. Such structures (with the lateral inhibition relayed through inhibitory inter-neurons) are ubiquitous in the brain [8].**

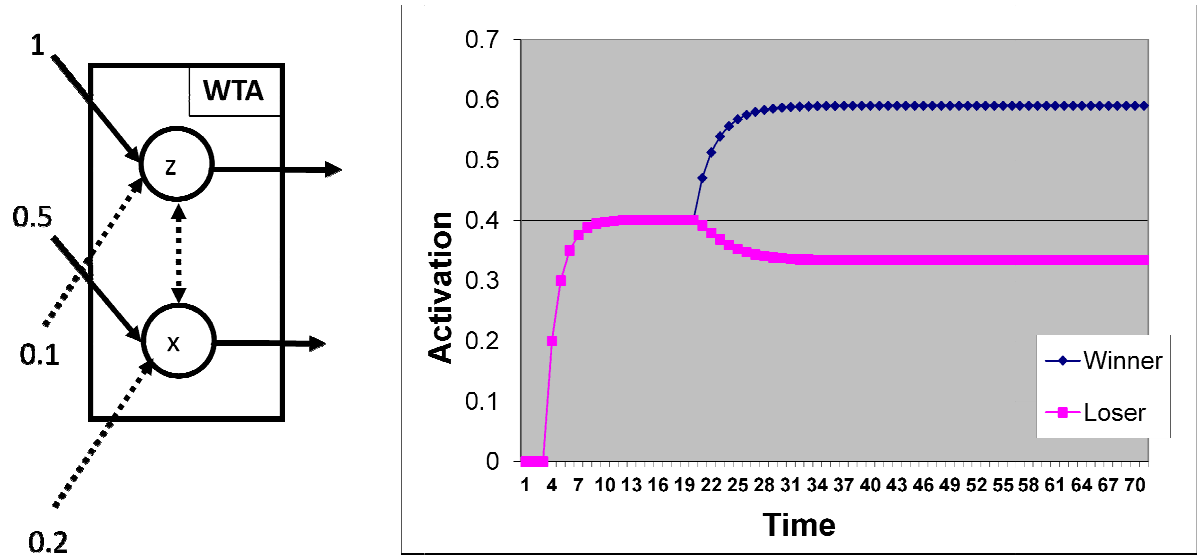


**Figure NET1: Left Panel: A winner-take-all network. Dashed arrows are inhibitory, unbroken arrows are excitatory. Right Panel: Activation dynamics of WTA given constant input, which begins at time step 20 and remains on from that moment. WTA unit Z is the winner (black line, with diamonds) and X the loser (grey line, with squares). The dynamics before time point 20 reflect prestimulus settling, under which Z and X attain their resting activation level. In this prestimulus settling period, winner and loser traces follow the same trajectory and are sat on top of one another.**

**Such a model might be used in a simulation of a stimulus response task. For example, the two WTA units would sit in the response pathway, with one corresponding to a left hand response and the other right. The lateral inhibition between responses (which is intrinsic to WTA) would reflect that only one response should be made at a time. The excitatory (extrinsic) input links might carry stimulus evidence for each response, e.g. an arrow stimulus pointing to the right excites the right response, etc. The inhibitory extrinsic input links might reflect a top-down bias, e.g. an experimental manipulation of the expectation of left versus right responses, with the less expected more inhibited. That is, the X response is less expected and thus is inhibited more. This is due to the 0.2 input, as opposed to 0.1.**

**Let us view this network, which we call WTA, as a component and assume that all excitatory input links (which are extrinsic) have the same weight, all inhibitory extrinsic input links have the same (negative) weights, and both lateral inhibition links (which are intrinsic) have the same (negative) weight. We also assume that all inputs are stable; that is they come on (at 20 cycles) and stay on (at the same level) until simulation end. The behaviour of this system is straightforward and quite as expected. The WTA unit that receives the highest net input (unit Z), i.e. sum of weighted inputs, wins the competition relatively soon after input starts. This is shown in the activation dynamics in the right panel of figure NET1.**

**Now, consider different ways of composing this WTA component with other components. Here, the four extrinsic links projecting into WTA (two excitatory and two inhibitory) are viewed as an implicit input interface and the two extrinsic output links (both excitatory) as an implicit output interface. A simple feed forward composition would be as in figure NET2.**

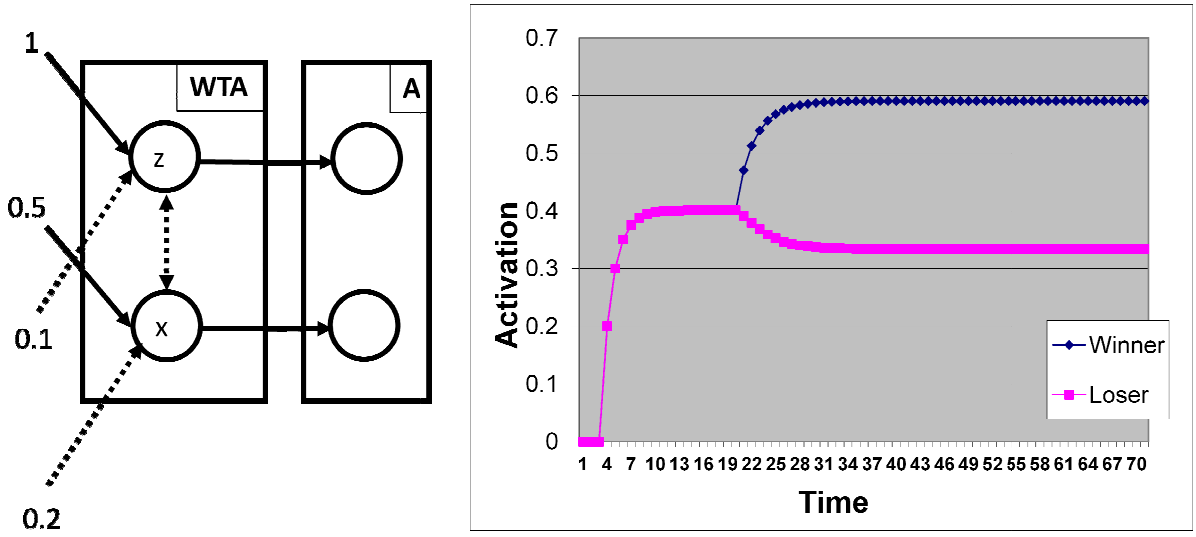


**Figure NET2: Two component response model, with a pure feedforward composition. WTA behaviour (shown in right panel) is unchanged. Again, the winner unit is Z (black lines, with diamonds) and the loser is X (grey lines, with squares).**

**Such a model might arise if component A was a next step in the response pathway. Importantly though, the addition of component A has no affect on the dynamics of component WTA. Hence, in this feedforward composition context, WTA behaviour**

**remains isolated and thus, encapsulated. However, consider the composition in figure NET3.**

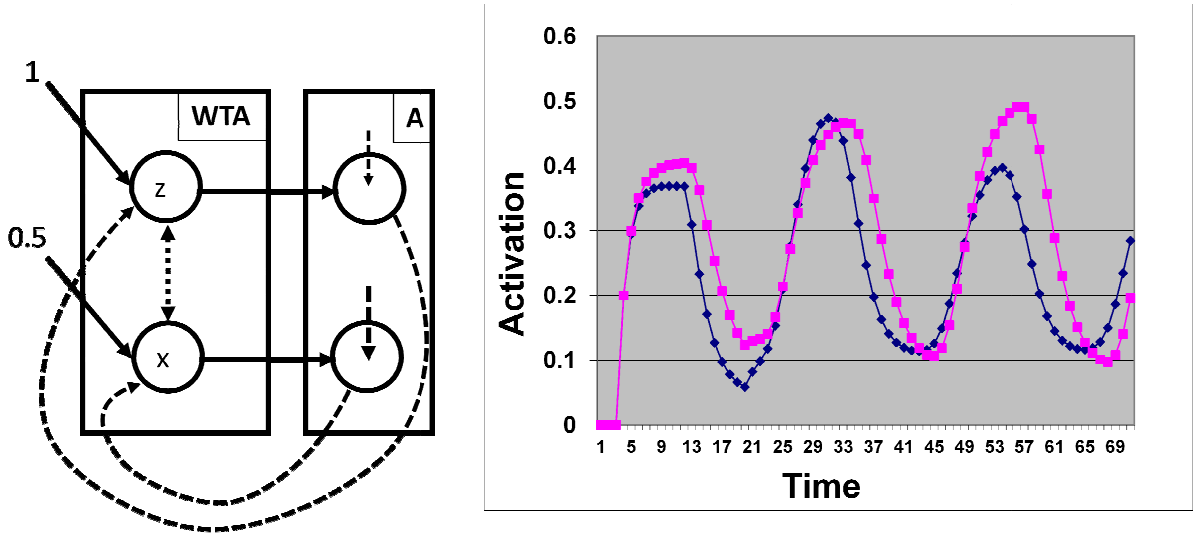


**Figure NET3: Two component response model, with a recurrent composition. WTA dynamics (right panel) are completely transformed. The activation of WTA unit Z is the black line with diamonds and the activation of WTA unit X is the grey line with squares. The units of component A have weak inhibitory biases, with strength indicated by point size of dashed vertical arrows. (Exact values of all weights and parameter settings are documented in the appendix.)**

**So, we have simply fed the output of component A units back to corresponding component WTA units, with the inhibitory weight on each link preserved (i.e. from when the link was an extrinsic input in the two component feedforward model). Thus, half of the WTA input interface has been bound to A's output interface. The resulting model, which is now of course recurrent, also corresponds to a very common structure in the brain. In particular, an ON unit exciting on OFF unit, which in turn inhibits the ON unit, occurs frequently in neurophysiologically detailed models. For example, such *opponent processes* arise embedded in the three laminar configuration of units in Dynamic Causal Models for EEG [36]. They also arise often in more cognitively-prescribed models, such as in the Simultaneous Type/ Serial Token model of temporal attention and working memory [35] and modelling of visual-motor control [34]. In particular, one might conceptualise the OFF nodes in figure NET3 as a mechanism to terminate ON node response preparation, when there is a break in feedforward evidence for that ON node, as discussed in [34].**

**But the critical point to note here is that, as shown in figure NET3 right panel, rewiring the inhibition into the WTA units, completely changes the dynamics of WTA. Effectively, "all bets are off" with regard to the behaviour of WTA; certainly, the emergent dynamics at WTA units Z and X are no longer obviously winner-take-all.**

**Although this example is certainly set-up to be a stark, indeed loaded, demonstration of nonencaspulability, the notion that a focal (in the sense of components) change in a neural network (here feeding links from A back to WTA) can effect a change in a remote component (here WTA), is certainly true. Furthermore, once recurrence is present, the perturbance in behaviour that a remote change might create could be dramatic, even threatening the very stability of the component.**

**It is important to note that step changes in neural dynamics can also be obtained without network restructuring. For example, without the addition of any extra components or the redirection of projections, which underlay the illustration of figures NET1, NET2 and NET3, we can regain stable winner-take-all dynamics from the figure NET3 network. Figure NET4 shows the activation dynamics that results from changing the biases in component A, i.e. increasing their inhibitory effect. Thus, a remote quantitative change, without qualitative (structural) change, can dramatically alter the dynamics of a component.**

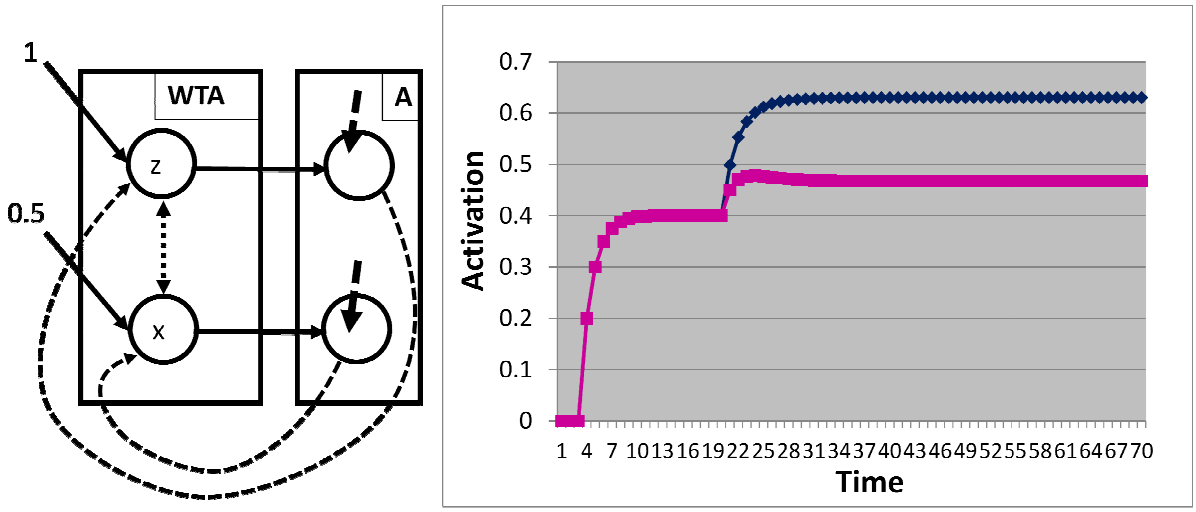


**Figure NET4: Two component response model (left panel), with a recurrent composition and strong inhibitory biases in component A. The dynamics of the WTA component (right panel) have returned to a predictable winner-take-all pattern. The line with black diamonds corresponds to unit Z and the one with grey squares to X.**

**Again, there is nothing in fact surprising here; suppressing the units of a component via strongly inhibitory biases, will clearly reduce that component's affect onto other components, which in our example breaks oscillatory couplings. This though once more merely emphasizes the fundamental argument: by their very nature, components of neural networks are "knitted together" and cannot obviously be isolated.**

**The implications of this absence of behavioural encapsulation would seem to be severe. In particular, incremental construction of large architectural-level neural models of mind and brain is likely to be extremely challenging, which may explain the dearth of such models and the emphasis on decompositionally small, highly specialised, models. We call this difficulty the *curse of (behaviour) decomposition flatness:* behaviourally, neural models (and particularly those with recurrence) cannot naturally be segregated into encapsulated components. In other words, typically they can only be considered at one compositionally and hierarchically undifferentiated level, i.e. by analogy with Petri Nets, they are, structurally flat.**

## Implications

**It seems then that, as formulated in neural networks, the nature of computation in the brain is substantially at odds with the "reductionism" required when constructing and considering architectural models of mind and brain. One consequence of this is that addition of components to such neural models is likely to mandate a complete refit of all model parameters. Effectively, behaviour can only be explored and calibrated globally. A familiar manifestation of this flatness is the difficulty in decomposing neural network learning. If one weight in a network is changed, it, at least potentially, affects every other weight and thus what has previously been learnt by a network. This irreducibility of weight change is not in fact surprising: one of the reasons for being interested in neural networks in the first place is that they are *global* constraint satisfiers. In neural networks, the properties a number of presynaptic neurons code can be simultaneously brought to bear on a single post-synaptic neuron; see for example, the joint influence of perceptual evidence and expectation (via extrinsic excitatory and inhibitory input respectively) on WTA response units (e.g. unit X) in our running example. That is, the activation of an**

**arbitrary unit is directly constrained by any unit that projects to it and indirectly by any unit that projects to a unit that projects to it, etc. If such global constraint satisfaction is sought, it is not surprising that component encapsulation is lost.**

**This decompositional flatness prompts a number of key questions.**

a) **How can one construct large, architectural-level, neural network models of cognitive function, as ultimately required of the cognitive neuroscience project?**
b) **Is the brain really (behaviour) decomposition flat? If not, how does the (structural) decomposition evident in the physical makeup of the brain provide behavioural encapsulation? Is there, for example, an effective interface-interlock between brain components, perhaps at the level of cortical columns, which protectively isolates component behaviour?**
c) **Is it even possible to formulate a calculus of communicating neural systems, which is meaningfully related to neural computation in the brain and, as the term calculus implies, exhibits any semblance of compositionality of behaviour?**

**In conclusion then, with respect to decompositional flatness, there does seem to be a conundrum at the heart of cognitive neuroscience research. Neuroscientists "informally" conceptualise the brain as possessing structural decomposition, psychologists "informally" conceptualise the mind as exhibiting the same characteristic, but the formal (behavioural) glue between the two – which is neural network modelling – is, it would seem, absolutely deficient in this respect.**

## Verifying Stability in Recurrent Neural Networks

**As our two component recurrent model suggests, the dynamics of recurrent neural networks can be extremely complex and, indeed, they can sit on the fringe of seeming chaos. In this context, network stability becomes a central concern; that is, can it be determined that a particular network will obtain a stable equilibrium activation pattern for a specific range of possible inputs? Identifying analytical solutions to such questions has proved difficult; indeed, it is likely that in the absence of gross simplifying assumptions, analytical approaches will prove intractable in many cases. It then becomes**

**interesting to consider whether the verification techniques used in concurrency theory, and formal methods in general, could be used to determine such stability. This is what we explore here.**

**Our focus will be on assessing the stability of learning algorithms for recurrent neural networks, particularly, neurophysiologically more realistic learning. Most theories of neural network learning are formulated for rather simple classes of networks and/ or under neurophysiologically rather unrealistic assumptions. As an example of the latter, the delta rule, and its generalisation, backpropagation [37], assume error values calculated at the output layer are relayed back through the network. This is required in order to adjust weights projecting into hidden layers. Biophysiologically, this suggests a signal that passes from post synaptic neurons, along dendritic trees, across synapses, down axons, back into presynaptic neurons, and all in the opposite direction to action potential transmission. Apart from through specialised pathways arising in specific brain structures, such as in regions of the cerebellum [8], signals that oppose the main electrochemical transmission direction have not been identified. Accordingly, more recent supervised learning proposals in cognitive neuroscience have focused on recurrent neural networks and posited that feedback activation from output towards input regions constrains hidden units according to the (clamped) teacher signal. That is, error is effectively fed back through activation, rather than a distinct error value.**

**A typical learning algorithm of this kind is the Generalised Recirculation (GenRec) algorithm [37], which is related to contrastive Hebbian learning. However, the reliance on feedback activation, amongst other aspects of neurophysiological plausibility, raises the possibility that the algorithm may not exhibit good convergence properties, i.e. learning might not be stable. In particular, our experiences with GenRec made us suspicious that it may suffer an especially severe form of catastrophic forgetting. It is well known that delta rule and backpropagation learning can forget what has previously been learnt when trained on a new pattern set. However, GenRec seems to be able to forget while still being trained on the *same* pattern set, i.e. a kind of spontaneous fixed-environment forgetting. Clearly, such an extreme propensity to unlearn would be a problem for both artificial and natural learning systems.**

**In response, we have sought to investigate the stability of GenRec learning using model checking. Note, although where possible analytical approaches have been used,**

**many evaluations of neural learning algorithms have relied upon simulation, basically repeatedly running the learning algorithm across different initial weight settings. Each run, though, must necessarily be terminated at some, effectively arbitrary, run length. At least in principle, model checking enables an exhaustive search of the learning space for a given initial weight setting. In this sense, the benefit of model checking over simulation in this context is similar to the traditional formal methods argument for verification over testing in software and systems engineering. That is, testing can reveal the presence of errors, but not certainty of their absence. Accordingly, simulation could reveal the presence of an instability in learning, but not certainty of its absence. For a given weight setting, model checking can support such an inference. We summarise our model checking of GenRec learning here, with full details presented in [38].**

**In order to enable us to use model checking, we first introduced a coding of neural networks in communicating automata. This involved a discrete neural update cycle, with activation transmitted across synchronous message passing channels, which effectively play the role of neural network links. Neural units were modelled as automata that update their membrane potential and output activation once the activation on all input links has been sampled. The resulting neural unit model is depicted in "pseudocode" in figure NEURON AUTOMATA.**

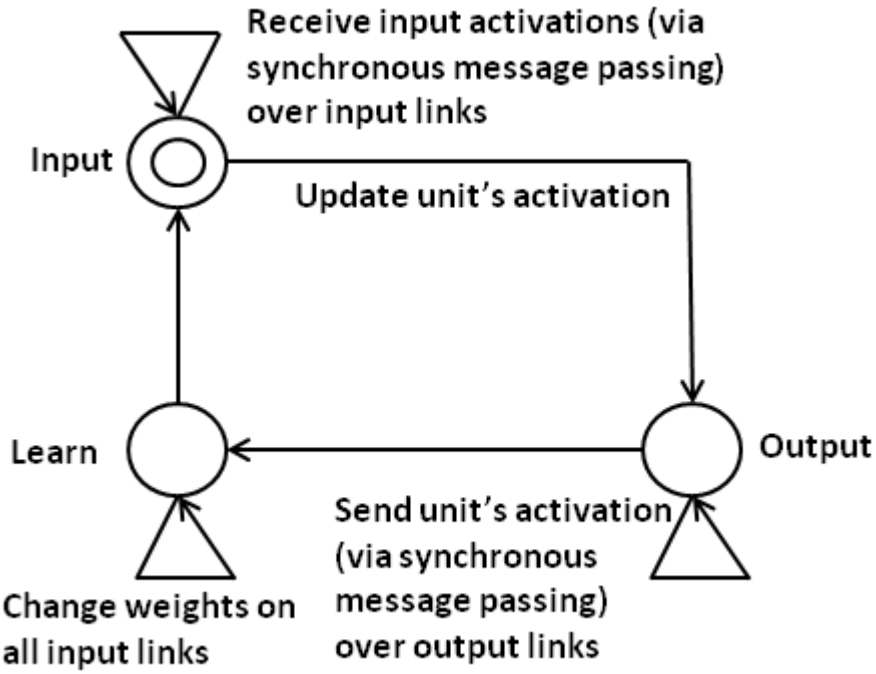


**Figure NEURON AUTOMATA: A neuron automaton**

**Thus, activations over links projecting to a unit are sampled at location Input. Using these input activations, the unit then updates its own activation (on the Input to Output link) by calculating excitatory and inhibitory input conductances, membrane potential and (rate coded) activation. This is followed by transmission of the unit's activation over output links at location Output and weight update across its input links at location Learn. The particular learning algorithm applied determines how weights change, but the neuron automaton structure remains the same.**

**Our communicating automata specification enforces a strict update cycle, which has a number of characteristics. The most important of these is that all our models, whether purely feedforward or recurrent, have an ordered series of layers, e.g. Input layer then Hidden layer then Output. We use the terms predecessor and successor to refer to this series, e.g. Input layer units are predecessors of hidden layer units. The specification updates unit activations according to this order. Thus, Input layer units are updated first, then initial (if there are many) hidden layer, and so on. However, importantly, if the network being specified is recurrent, both predecessor and successor units can contribute to the activations received at a neuron automaton's Input location. At this point though, only the predecessor units will themselves have been updated under the current cycle. As a consequence, each unit is updated according to the current activation of predecessor units and previous cycle activation of successor units.**

**Our specifications involved a composition of three layers of such unit automata with an environment automaton, which instantiated patterns on input and output layers and calculated learning errors. In this framework, we implemented classic Backpropagation (with purely feed forward activation), Backpropagation over a recurrent network (but with error still propagated independently of activation) and the Generalised Recirculation algorithm.**

**We then recast the network into PROMELA, the implementation language of the SPIN linear-time model checker [39]. The three PROMELA models (BP, BPRec and GenRec) were then instantiated to learn the XOR problem, which is a small, in terms of number of units and patterns, but canonically hard learning problem. One might indeed call it the Dining Philosophers of neural network learning problems!**

**We were interested in three correctness properties, each of which we formulated in linear-time temporal logic. These formulae refer to a state property, SUCCESS, which holds when the activation at each output layer unit is within 0.5 of the desired activation (given by the teacher pattern). The first formula, \Diamond \Box SUCCESS, expresses "stability", meaning the system eventually (i.e. \Diamond) reaches a state from which SUCCESS will always remain (i.e. \Box) true. The second formula, \Box \Diamond SUCCESS, expresses "recurrence", meaning it will always be the case that a SUCCESS state can eventually be reached. The third, \Diamond SUCCESS, expresses just "eventuality", meaning that from the start state, SUCCESS will eventually be reached. Clearly, stability strictly implies recurrence, which strictly implies eventuality.**

**We model checked a range of initial weight settings. As shown in table RESULTS, GenRec did indeed perform very poorly on the stability property. Follow-up verifications (presented in [38]) demonstrated that GenRec was more likely to satisfy stability as the sigmoidal function became shallower, i.e. the gain parameter was reduced, making the function closer to linear. Thus, as neurons become more discriminating, in the sense that differences in input strength are reflected in similar sized differences in output activation, their training becomes more stable in a fixed environment.**

| $\gamma = 50$ | Initial Weights | Stability | Recurrence | Eventuality |
|---|---|---|---|---|
| BP | 0-0.2 | 100% | 100% | 100% |
| | 0.2-0.4 | 100% | 100% | 100% |
| | 0.4-0.6 | 100% | 100% | 100% |
| | 0.6-0.8 | 0 | 0 | 0 |
| | 0.8-1 | 0 | 0 | 0 |
| BPRec | 0-0.2 | 100% | 100% | 100% |
| | 0.2-0.4 | 100% | 100% | 100% |
| | 0.4-0.6 | 100% | 100% | 100% |
| | 0.6-0.8 | 100% | 100% | 100% |
| | 0.8-1 | 100% | 100% | 100% |
| GeneRec | 0-0.2 | 17% | 100% | 100% |
| | 0.2-0.4 | 0 | 100% | 100% |
| | 0.4-0.6 | 0 | 100% | 100% |
| | 0.6-0.8 | 0 | 100% | 100% |
| | 0.8-1 | 0 | 100% | 100% |

**Figure RESULTS: Model checking results. Initial weights were sampled at random from the highlighted range and \gamma denotes the sigmoidal's gain. As can be seen, GenRec was very unstable in its learning of XOR.**

## Conclusion

**We have considered the potential for concurrency theory to contribute to modelling and analysis in cognitive neuroscience. The (behavioural) decomposition flatness of neural networks and verification of the (environment-fixed) stability of learning algorithms were explored.**

**Although not considered here, we also believe that concurrency theory abstraction techniques based on reduction of nondeterminism could be applied in cognitive neuroscience; see our discussion of this topic in [40]. Such methods could plausibly contribute to addressing the Irrelevant Specification problem, as formulated, for example, by [3].**

**Our main conclusions are, though, threefold. Firstly, it does seem that decomposition flatness of neural networks is a barrier to formulating architectural models of mind and brain in cognitive neuroscience. Secondly, there is potential in using model checking to determine the stability properties of neurophysiologically prescribed neural models. Thirdly, it may be that the GenRec learning algorithm exhibits a particularly severe form of instability, in which the system forgets while learning in a fixed environment.**

## Appendix

**We document the activation equations and parameter settings used in our illustration of nonencapsulability; see section "Decomposition and Neural Behaviour". The basic activation equations are as follows. For an arbitrary unit *i*, the net input, i.e. sum of weighted input activation, is given by,**

$$\boldsymbol{net_i(t)} = \sum_{\boldsymbol{j \in To(i)}} \boldsymbol{w_{ji}}\ \boldsymbol{a_j(t)}$$

**where, *t* denotes discrete time, i.e. update cycles, *To(i)* is the set of units that project to unit *i*, $w_{ji}$ is the weight between units *j* and *i* and $a_j(t)$ is the current (output) activation at**

**unit *j*. A unit's activation is then calculated through time averaging of the logistic transformed net input, i.e.**

$$\boldsymbol{a_i(t) = \tau . \sigma\big(net_i(t-1)\big) + (\tau - 1). a_i(t-1)}$$

**where, $\tau$ is a time constant (taking a value between zero and one) governing the rate of change, and $\sigma$ is the standard, sigmoidal shaped, logistic function, i.e.**

$$\boldsymbol{\sigma(x) = \frac{1}{1 + e^{-g(x+b)}}}$$

**where, *g* is the gain, i.e. steepness of the logistic function, and *b* is a constant bias.**

**<u>Parameter settings</u>**

**The parameter and weight settings of the neural models discussed in section "Decomposition and Neural Behaviour", c.f. figures NET1, NET2, NET3 and NET4, are documented here. We use the following terminology: in layer A, we call the unit that Z projects to A.top and the unit X projects to A.bottom.**

**<u>Parameters fixed across all models</u>**

**Time constant ($\tau$): 0.4**

**Sigmoidal gain (*g*) at Z and at X: 1**

**Bias (*b*) at Z and at X: 0**

**<u>Weights fixed across all models</u>**

**Extrinsic Excitatory to Z and to X: 1**

**Extrinsic Excitatory from Z and from X (to A.top respectively A.bottom in NET2, NET3 and NET4): 6**

**Inhibitory Intrinsic from Z to X and X to Z: -1**

**<u>Parameters Varying Across Models</u>**

**Biases (*b*) in Layer A**

**Net 2 and Net 3 A.top: -1**

**Net2 and Net3 A.bottom: -2**

**Net4 A.top and A.bottom: -10**

**Sigmoidal gain (*g*) at A.top and A.bottom (NET2, NET3 and NET4): 3**

**Extrinsic Inhibitory**

**Net1 and Net2 inhibitory extrinsic to Z and to X: -3**

**Net3 and Net4 inhibitory from A.top to Z and A.bottom to X: -3**

**References**


1. Johnson-Laird, P., *Mental Models*. 1983, Cambridge, Massachusetts: Harvard University Press.
2. Fodor, J.A., Pylyshyn, Z.W., *Connectionism and cognitive architecture: a critical analysis.* Cognition, 1988. 28: p. 3-71.
3. Newell, A., *Unified Theories of Cognition*. 1990, London: Harvard University Press.
4. Penrose, R., *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*. 1999, Oxford: Oxford University Press.
5. Dennett, D., *Consciousness explained*. 1991, London: Penguin.
6. Miller, G., *The cognitive revolution: a historical perspective.* TRENDS in Cognitive Sciences, 2003. 7(3): p. 141-144.
7. Haykin, S., *Neural Networks and Learning Machines (3rd Ed.)*. 2009, New York: Prentice Hall.
8. Rolls, T.E. and A. Treves, *Neural Networks and Brain Function*. 1998, Oxford: Oxford University Press.
9. Roscoe, A.W., *The Theory and Practice of Concurrency*. 1998, Harlow: Pearson Education.
10. Bowman, H., Gomez, R., *Concurrency Theory: Calculi and Automata for Modelling Untimed and Timed Concurrent Systems*. 2006, London: Springer.
11. Bowman, H. and J. Derrick, eds. *Formal Methods for Distributed Processing*. 2001, Cambridge University Press: Cambridge
12. Barnard, P.J., *Interacting Cognitive Subsystems: modelling working memory phenomena within a multi-processor architecture.*, in *Models of Working Memory: Mechanisms of active maintenance and executive control*. 1999. p. 298-339.
13. Baddeley, A.D., Logie, R.H., *Working Memory: The Multiple-Component Model*, in *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, A. Miyake, Shah, P., Editor. 1999, Cambridge University Press: Cambridge.

14. **Gazzaniga, M.S., R.B. Ivry, and G.R. Mangun, *Cognitive Neuroscience: the Biology of the Mind*. 1998, London: Norton.**
15. **Rumelhart, D.E., J.L. McClelland, and the-PDP-Research-Group, *Parallel Distributed Processing, Explorations in the Microstructure of Cognition. Volume 1: Foundations and Volume 2: Psychological and Biological Models*. 1986, Cambridge, Massachusetts: A Bradford Book, The MIT Press.**
16. **Li, Z., *Computational design and nonlinear dynamics of a recurrent network model of the primary visual cortex.* Neural Computation, 2001. 13(8): p. 1749-1780.**
17. **Milner, R., *Process Constructors and Interpretations.* Information Processing, 1986. 86: p. 507-518.**
18. **Milner, R., *Communication and Concurrency*. 1989: Prentice-Hall.**
19. **Hoare, C.A.R., *Communicating Sequential Processes*. 1985, London: Prentice Hall.**
20. **Derrick, J., et al. *Comparing LOTOS and Z refinement relations*. in *FORTE/PSTV'96*. 1996. Kaiserslautern, Germany: Chapman & Hall.**
21. **Bowman, H., et al., *A formal framework for viewpoint consistency.* Formal Methods in System Design, 2002. 21(2): p. 111-166.**
22. **Bowman, H., et al., *Strategies for consistency checking based on unification.* Science of Computer Programming, 1999. 33: p. 261-298.**
23. **Bolognesi, T., Brinksma, E., *Introduction to the ISO specification language LOTOS.* Computer Networks and ISDN Systems, 1988. 14(1): p. 25-29.**
24. **Bergstra, J.A. and K. J.W., *Algebra for Communicating Processes with Abstraction.* Journal of Theoretical Computer Science, 1985. 3: p. 77-121.**
25. **Bowman, H., Faconti, G., *Analysing cognitive behaviour using lotos and mexitl.* Formal Aspects of Computing, 1999. 11: p. 132-159.**
26. **L. Su, B., H., Barnard, P.J., Wyble,B., *Process algebraic modelling of attentional capture and human electrophysiology in interactive systems.* Formal Aspects of Computing, 2009. 21: p. 513-539.**
27. **Bowman, H., Su, L., Barnard, P.J., *Salience Sensitive Control, Temporal Attention and Stimulus-rich Reactive Interfaces*, in *Human Attention in Digital Environments*, C. Roda, Editor. 2011, Cambridge University Press.**
28. **Anderson, J., Lebiere, C. , *The Atomic Components of Thought*. 1998, Mahwah: Laurence Erlbaum.**
29. **Kieras, D.E., Meyer, D.E., Mueller, S., Seymour, T. , *Insights Into Working Memory from the Perspective of the EPIC Architetcture for Modelling Skilled Percpetual-Motor and Cognitive Human Performance*, in *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, A. Miyake, Shah, P., Editor. 1990, Cambridge University Press: Cambridge.**
30. **Hubel, D.H. and T. Wiesel, *The Ferrier Lecture: Functional architecture in macacue monkey visual cortex.* Proceedings for the Royal Academy, London, Series, B, 1977. 198: p. 1-59.**
31. **Anderson, A.K. and E.A. Phelps, *Lesions of the human amygdala impair enhanced perception of emotionally salient events.* Nature, 2001. 411(6835): p. 305-9.**
32. **Anderson, J.R., Fincham, J. M. Qin, Y., Stocco, A., *A central circuit of the mind.* Trends in Cognitive Sciences, 2008. 12(4): p. 136-143.**
33. **Machens, C.K., Romo, R. Brody, C. D., *Flexible Control of Mutual Inhibition: A Neural Model of Two-Interval Discrimination.* Science, 2008. 307(5712): p. 1121-1124**
34. **Bowman, H., Schlaghecken, F., Eimer, M., *A neural network model of inhibitory processing in subliminal priming.* Visual Cognition, 2006. 13(4): p. 401-480.**
35. **Bowman, H. and B. Wyble, *The Simultaneous Type, Serial Token Model of Temporal Attention and Working Memory.* Psychological Review, 2007. 114(1): p. 38-70.**

**36.** **David, O., Kiebel, S. J., Harrison, L.M., Mattout, J., Kilner, J.M., Friston, K.J., *Dynamic Causal Modeling of Evoked Responses in EEG and MEG.* NeuroImage, 2006. 30(4): p. 1255-1272**

**37.** **O'Reilly, R.C. and Y. Munakata, *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain.* 2000: MIT Press.**

**38.** **Li, S., Gomez, R., Bowman, H., *Analysing Neurobiological Learning Using Communicating Automata.* (Under submission), 2011**

**39.** **Holzmann, G.J., *The Model Checker SPIN.* IEEE Transactions On Software Engineering, 1997. 23(5): p. 279-95.**

**40.** **Barnard, P.J. and H. Bowman, *Rendering information processing models of cognition and affect computationally explicit: Distributed executive control and the deployment of attention.* Cognitive Science Quarterly, 2004. 3(3): p. 297-328.**