UC Davis
UC Davis Previously Published Works

Title

Connectionist Mechanisms for Cognitive Control

Permalink
https://escholarship.org/uc/item/5h75g3tf

Journal
Neurocomputing, 65

ISSN
0925-2312

Authors
Wendelken, Carter
Shastri, Lokendra

Publication Date
2005-06-01

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Connectionist Mechanisms for Cognitive Control

Carter Wendelken, Lokendra Shastri

Abstract

An understanding of cognitive control is crucial for understanding high-level cognition and delineating the functional role of prefrontal cortex in supporting complex cognitive operations. In this paper, we approach the problem of cognitive control by examining the control needs of SHRUTI, a neurally plausible and cognitively motivated model of inference and decision-making. It is shown that processing via spreading activation has a number of limitations with respect to inference and decision-making, and specific forms of controlled processing is required to overcome these limitations. We propose a set of primitive, neurally plausible control mechanisms, including monitoring, filtering, selection, maintenance, organization, and manipulation, describe connectionist implementations of these primitive mechanisms, and demonstrate the use of several of these primitives in a complex control process.

1. Introduction

Central to the understanding of many complex cognitive operations, and of the nature of brain regions such as the prefrontal cortex, is an understanding of cognitive control – what is cognitive control, what is it involved with, and how is it realized? In this paper, we approach the problem of cognitive control by analyzing the control needs of SHRUTI, a neurally plausible and cognitively motivated model of inference and decision-making [1-4]. Based on this analysis, we propose a set of neurally plausible connectionist control mechanisms and demonstrate their use as primitives in a complex control process.

Cognitive control, or executive control, refers to a set of neural cognitive processes typically involved in the execution of complex tasks and most often associated with activity of prefrontal cortex. Capacities that are frequently included under the rubric of cognitive control include attention [5], conflict detection [6], working memory maintenance [7], working memory manipulation [8], and task-switching or dual-task performance [9], among others. Cognitive control is thought to be especially important in situations where tasks are novel or unlearned [10]. The connection between cognitive control and prefrontal cortex is well-established, through imaging (e.g. [11]), single-cell-recording (e.g. [7]), primate lesion studies (e.g. [8]), and neuropsychology (e.g. [9]). Importantly, prefrontal cortex is characterized by a significant level of connectivity to many other brain regions, particularly those involved in high-level, multi-modal representation [12].

2. The SHRUTI model of inference and decision-making

The immediate motivation for this work was the need for control mechanisms that became apparent as SHRUTI, a model of reflexive reasoning via spreading activation, was transformed into a decision-making architecture (SHRUTI-agent). A basic description of this model, and the limitations that make control necessary in some situations, is provided here.

2.1. Principal features of the SHRUTI model

SHRUTI is a structured connectionist model wherein entities, categories, rules, and relations are all represented via *focal clusters*, or structured collections of nodes with different nodes subserving different functions. For example, the systematic knowledge (rule) "if someone buys something, then they own it" is represented by a focal cluster for the relational predicate (or action) "buy", another focal cluster for the relational predicate "own", and a third focal cluster for the rule that is used to link these two relations together (see Figure 1). Nodes in SHRUTI are meant to correspond to small ensembles of neurons; they are capable of maintaining a limited amount of state information and may have a range of different activation functions. Node types seen in Figure 1 include the collector ("+"), whose activity indicates affirmation or belief that something is true, the enabler ("?"), whose activity indicates a query, the utility node ("$"), whose activity indicates reward or punishment associated with the relation, and role nodes and type nodes, whose activity specifies relational role fillers. An important feature of the SHRUTI model is its use of temporal synchrony as the mechanism to achieve variable binding. Specifically, synchronous activity of a role node (e.g. "buys-buyer") and a type or entity node (e.g. "Tom") indicates that type as the filler of that role ("Tom is the buyer").

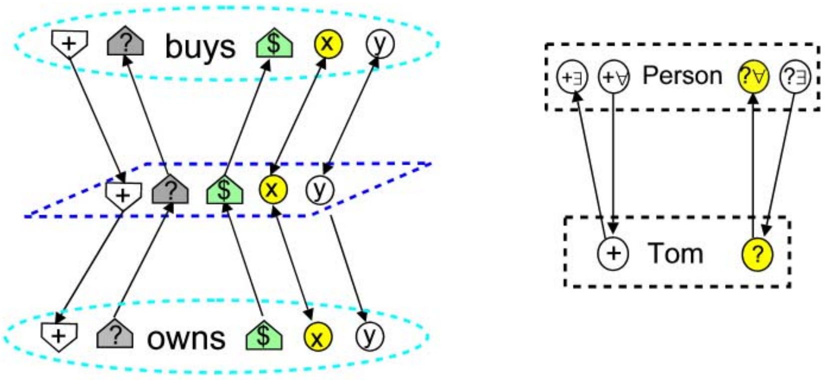

Figure 1:

Focal cluster encoding of a causal rule and a simple type hierarchy. Shading indicates node activation. Shown is the transformation, via spreading activation, of the query “Does Tom own something?” into the query “Did Tom (or did everyone) buy something?”.

2.2. Inference and decision-making in SHRUTI

Inference in SHRUTI is initiated with the assertion of some relational instance (e.g. “Tom bought a car”) or query (e.g. “Who owns the bike?”) via activation of appropriate nodes, and it proceeds as spreading activation across connected focal clusters. Specifically, for a causal rule with one or more antecedent relational clusters and one or more consequent relational clusters, activity propagates from collector to collector in the forward (antecedent to consequent) direction and from enabler to enabler in the backward direction. Synchronized phasic activity propagates in either direction from role nodes on one side of the rule to role nodes on the other side. Results of inference can be extracted from the network state at any point in time. The time to obtain complete inferential results is proportional to the depth of relevant causal chains and is independent of the total size of the knowledge base.

The presence of utility nodes and links and of action focal clusters, in addition those elements used for inference, allows decision-making behavior to emerge. Reward or punishment associated with a relational instance is propagated from utility node to utility node backwards (in the consequent to antecedent direction) along a causal chain. In the example in Figure ??, the desirability of ownership causes the “buy” action to be also marked as desirable. In general, utility propagates from goals to subgoals, and designation of some relational instance as a goal can lead ultimately to utility node activation at one or more actions (which sit at the tops of some causal chains), and hence to the ability to make a decision about what action to execute.

Of course, the whole story is significantly more complicated than this. Correct propagation of utility requires consideration not only of the utility at a relation, but also of beliefs in neighboring relations.
Actions require special consideration, since unlike other relations they are under the direct control of the agent.
Combination of rewards or punishments from different sources must be carefully controlled.
These and other modifications to the basic spreading activation model, that together enable effective reactive decision-making, are described in detail elsewhere [3, 4]. But even with this range of modifications, the spreading activation model is not sufficient to handle a large class of complex decision problems. For this we must look to different sorts of connectionist mechanisms.

2.3. Limitations of spreading activation

SHRUTI as described so far is a model of processing without control; all operations occur via spreading activation over a representational substrate and automatically as a consequence of that representational structure. The implementation of decision-making via uncontrolled processing has a number of limitations that are inherent to any spreading activation-based model. For instance, the mechanism of temporal-synchrony variable binding results in a soft bound on the number of entities that can be represented by the system. This means that an episode of complex decision-making, and complex inference in general, that involves a large number of entities must be broken down into simpler segments. The spreading activation mechanisms do not, on their own, produce this requisite segmentation. There is a high structural and computational cost associated with multiple instantiation of relational predicates. This means that the number of active instantiations of any given relation must be kept to a small number during an episode of reasoning or decision-making. This too can necessitate breaking a complex episode into smaller elements. An expected limiting factor for a neural realization of the SHRUTI model is the imprecision of the machinery that propagates synchronous activity. Although this is not a problem for a connectionist implementation, this lack of precision means that as the depth of propagation increases, coherence decreases, eventually leading to a loss of binding information. When extensive propagation is required to solve a problem, some mechanism must cope with this potential loss of information. A particularly important limitation of the spreading activation model as it applies to sequential decision-making is its lack of an explicit representation of time. While the system can predict the consequences of a single action, it often fails to predict the result of a sequence of interdependent actions, since it might not be possible to simultaneously instantiate the network with the conditions that hold prior to performance of the first action (necessary for predicting the consequences of the first action) and the conditions that hold prior to performance of the second (necessary in order to predict its consequences).

3. Cognitive control

Mechanisms of cognitive control can help to overcome the limitations of the spreading activation model in the context of decision-making and other complex cognitive tasks. The goal here, and one that is shared by other researchers (e.g. [13, 14]), is to develop detailed connectionist models of neural mechanisms that seem critical for supporting various cognitive control tasks. A prerequisite to exploring how control can aid decision-making and other cognitive tasks is to have a workable and sufficiently concrete definition of control. But what does it mean to say that something is a control mechanism?

From a significant amount of work in psychology, neuropsychology, and cognitive neuroscience, it has become clear that there is a system or set of mechanisms within the brain, primarily based in the prefrontal cortex, that are characterized by flexibility and the capacity to operate on a wide range of lower-level representations, and that serve to systematically manipulate or influence some aspect of cognition. In connectionist terms, we define control primarily in contrast to the representation-bound spreading activation that characterizes the operation of the SHRUTI model. Control mechanisms are not generally bound to a particular representational substrate; instead a single control circuit might operate on a wide range of different contents. They are not activated automatically whenever some content is accessed. They typically operate in a serial manner, or with very limited capacities, and do not scale up in the way that spreading activation can. A control mechanism serves to manipulate the underlying spreading activation in some manner. It might cause some process to repeat or iterate. It might maintain activation that would otherwise decay. It may block or enhance activation of select nodes, node types, regions, or pathways.

Controlled processing may appear quite complex, but it need not require a particularly complex central controller. In fact, control may be highly distributed, and computationally quite simple. The addition of a few simple control mechanisms, each adding to the already significant capability of the underlying spreading activation system, will prove adequate to deal with many complex cognitive tasks.

4. Connectionist control mechanisms

In transforming SHRUTI into a decision-making agent architecture, two sorts of control mechanisms were considered: complex control mechanisms and control primitives. Complex control mechanisms directly address the requirements for decision-making. They enable the system to do such things as simulate sequences of actions, focus on particular subgoals, test action or predicate hypotheses, control search via uncertainty monitoring, and maintain and manipulate item sets (see [4] for a complete description). Each complex control mechanism directly addresses some failure of the spreading activation model. For example, the subgoal focus mechanism allows the system to restrict its operation to a particular sub-network, in order to get past fundamental limits such as the number of active entities. The hypothesis-testing mechanism allows the system to cope with complex decision scenarios where early execution of one requisite action prevents later execution of another. In these circumstances, propagation of utility in SHRUTI can essentially become deadlocked, and the decision process can reach a premature dead end, without appropriate intervention. Search control provides the system with the capacity to overcome depth limits on search at points in the network where it is most important to do so. This allows the system to solve problems that are beyond the capacity of spreading activation alone due to the depth of associated causal chains. We posit that these complex mechanisms are in turn built from a common set of simple connectionist control primitives. Based on the computational needs of the complex control mechanisms, control primitives that accomplish monitoring, filtering, selection, maintenance, organization, and manipulation were developed. Following a description of each control primitive and its connectionist implementation, one illustrative case, the utilization of subgoal focusing in an implemented military decision-making scenario, is presented.

4.1. Monitoring

Monitoring is one of the simplest control primitives. It involves obtaining input from a set of targets and signaling whenever some specific property is satisfied by the input set. Detecting a simple property such as “all nodes are firing above some threshold” can be achieved with a single monitor node; complex input properties are more easily detected by a set of nodes or circuit. Monitors are particularly useful for detecting properties that are not local to any one part of the network.

An important form of monitoring is conflict detection. A conflict node in SHRUTI obtains input from both positive and negative collectors of a predicate focal cluster, and becomes active whenever the product of their activity is sufficiently high, signaling conflict at that predicate (i.e., the occurrence of both, a high level of belief and a high level of disbelief, about a relational instance) [15]. At a higher level, another monitor node receives input from many such conflict nodes and becomes active when conflict is found anywhere in the system. A single high-level conflict detection node, connected to local (focal-cluster specific) conflict detection nodes, is a simple monitoring circuit currently implemented in SHRUTI (see Figure 2a).

4.2. Filtering

Another basic aspect of control is the ability to filter, or selectively enhance or suppress activity in some part of the network. A filter operating on a SHRUTI network might suppress the activity of a set of goal nodes. Another filter might enhance a set of queries. A third filter mechanism could entirely block the operation of one causal subnetwork to prevent it from interfering with the propagation of activity in another.

Filters have a particularly simple connectionist implementation, as shown in Figure 2b. Any node that has excitatory or inhibitory connections to a relatively large set of nodes, or one that has inhibitory projections to nodes in one set and excitatory projections to nodes in another set is effectively a simple filter. Appropriate combinations of simple filters produce useful complex filters that enhance certain aspects of a representation or computation, and suppress others.

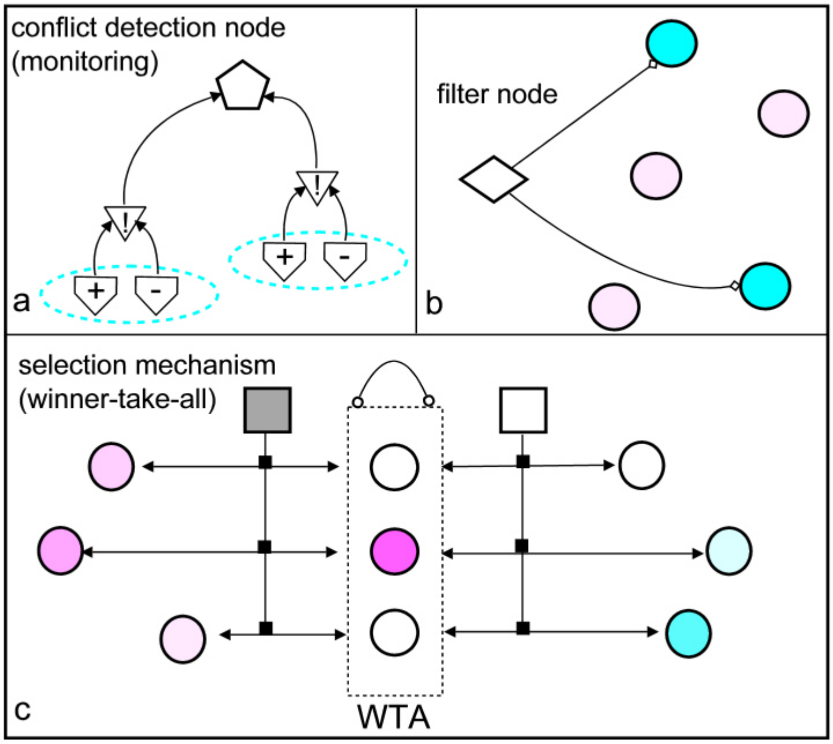

Figure 2:

Connectionist implementations of various control primitives.

(a) A global conflict detection node receives input from local conflict detectors within predicate focal-clusters.

(b) A filter node selectively enhances activation in a set of target nodes.

(c) A winner-take-all selection mechanism is connected to two separate sets of nodes, with interaction between the WTA and set on the left side currently enabled.

4.3. Selection

Another capacity that is important for a number of different control processes is the ability to select a node for further processing from some target set. The selection might be random or it might be based on activity level. For example, a common task might be to choose the most active node from a set. Whatever selection strategy is chosen, the selection mechanism must have a way to impose its choice on subsequent processing.

A winner-take-all (WTA) network [16] is a kind of selection mechanism. In a WTA, a set of nodes with inhibitory interconnections competes until only one node remains active. To serve as a control mechanism, a WTA can be connected to, but be physically distinct from, the nodes on which it operates. Some trigger allows activity to flow to the WTA structure and, once the selection has been made, from the WTA back to the source nodes. A more general purpose WTA mechanism can be constructed by combining the simple WTA controller with a filter (see Figure 2c). Each WTA node is then connected to many source nodes but the filter only allows one set to communicate with the WTA mechanism at any given time.

4.4. Maintenance, Organization, and Manipulation

The nodes used in SHRUTI to represent types, entities, relations, and rules are active only in response to input, and their activity decays quickly when that input disappears. Sometimes it is useful to maintain activity at a node or set of nodes beyond its normal time-span. Maintenance mechanisms accomplish this goal. Importantly, a maintenance mechanism does not create new activity at a node but instead sustains activity that was introduced from some other source (in effect, such a maintenance mechanism serves as a form of short-tem memory).

Maintenance is achieved in SHRUTI with a very simple connectionist implementation, involving, minimally, a single link from a maintainer node to a target node. Maintainer nodes are activated (clamped) whenever there is a need to sustain activation. When inactive, the target node is not responsive to maintainer node activity. However, if the target node is active and at the same time receives input from the maintainer node, then the target node enters a new "maintained" state in which it becomes responsive to maintainer activity, even in the absence of other input. In the maintained state, the target node's level of activity corresponds to the activity level it had prior to the maintainer input. When the maintainer input disappears, the target node reverts to a "normal" state where its continued activation is dependent on other input. Thus, while the maintainer node associated with some target node is active, any otherwise short-lived activity that the target node acquires is sustained.

Although maintenance has been described in terms of the interaction between two nodes, it is possible that a single maintainer node could be responsible for many target nodes, and also that a single target node could have multiple maintainers. Structured much like a filter mechanism, a maintainer circuit would be associated with a particular set of targets, possibly very many or a specific few. For example, one maintenance mechanism could maintain activity at all action focal clusters, while another might maintain activity at entity query nodes for entities that are subtypes of *Person*.

The capacity to organize elements is critical for many tasks. Organization is related to maintenance, insofar as it is often useful to organize elements that are being maintained. Moreover, the relatively simple and local control structures involved in maintaining representations can be more accessible to an organizing control mechanism than the complex and distributed representations themselves. Manipulation is in some sense an extension of organization, in that manipulation typically involves re-organization of maintained elements. This is potentially a very broad category of control mechanisms. Simple manipulation operators might be a circuit that causes an ordered set of maintained elements to shift forward one place, or one that causes an arbitrary shuffling of targets. An implemented connectionist mechanism that achieves organization and manipulation of items in SHRUTI is described elsewhere (see [4]). Together, maintenance, organization, and manipulation mechanisms comprise what is commonly referred to as working memory. In earlier work, we present an account of how these functions might be implemented in lateral prefrontal cortex [17].

5. Operation of subgoal focus in an implemented decision-making scenario

In the decision-making domain, an ability to focus on particular goals or subgoals, while ignoring others, makes it possible to cope with complex situations that might otherwise be unsolvable. Subgoal focus refers to the complex control mechanism that makes this possible. An implemented scenario, wherein a simulated military commander must decide how to best preserve his unit, demonstrates the utility of the subgoal focus mechanism. (Network, knowledge base, and other implementation details are provided in [4]). Destroying the enemy, suppressing its fire, and managing to avoid contact are all viable options for the commander, and thus potential subgoals. Inference via simple spreading activation examines these subgoals simultaneously. But consideration of how to attack the enemy force involves a different set of entities and locations than does consideration of how to avoid contact, so trying to process both options at the same time leads to an overtaxing of the system – in particular to an inability to activate all relevant entities. When the subgoal focus mechanism is inserted, this mechanism monitors the "complexity status" – here number of active entities – of ongoing inference and selectively blocks activation of all but the highest priority goal when that complexity becomes too high. In this case, as inference fails due to the activation of too many entities, activity at the subgoal predicates representing "suppress the enemy force" and "avoid contact with the enemy force" is inhibited so that the subgoal "destroy the enemy force" can be more fully explored.

Subgoal focus depends primarily on the monitoring, filtering, and selection control primitives. A monitoring mechanism is required first to detect those situations -- such as overuse of phases and relational predicate banks -- in which focusing on a subgoal might be useful. To detect the number of active entities, important in the above scenario, the mechanism has a monitor node connected to each element in the type hierarchy, that fires whenever it receives input in every phase (reflecting full use of available phases within a cycle.) When the subgoal focus mechanism is activated, a winner-take-all mechanism connected to predicate utility nodes is used to provide automated subgoal selection. When *destroy(EnemyForce)* is associated with the highest utility among the active subgoals (as happens here), then its proxy representation within the WTA controller wins out, it in turn is reinforced, and other subgoals are inhibited. Focusing on this single subgoal involves activating a filter that has a pre-wired association with the *destroy* predicate such that it enhances activity at this predicate and inhibits collector and enabler nodes on unconnected causal pathways. As an extension to this, sequential examination of multiple subgoals may be supported by the addition of a second filter that blocks activity at recently examined subgoals during a later iteration of this process. Thus through the combined operation of three control primitives – monitoring, selection, and filtering – the complex process of focusing on a particular subgoal or subgoals is achieved, and the system is able to make a decision under circumstances where spreading activation in the absence of control proved insufficient.

6. Conclusion

We have demonstrated that cognitive control can be achieved via relatively simple and neurobiologically plausible connectionist mechanisms. The set of control mechanisms presented here specifically reflects the needs of the SHRUTI decision-making architecture, allowing the system to overcome limitations that are inherent to its basic spreading activation mode of operation in the performance of complex decision-making tasks. While it is unlikely that this represents the complete set of control mechanisms that might exist in the brain, we expect that this understanding of cognitive control from a connectionist perspective will promote and inform neuroscientific research aimed at uncovering the neural basis of cognitive control.

References

1. Shastri, L. and V. Ajjanagadde, *From simple associations to systematic reasoning.* Behavioral and Brain Sciences, 1993. **16**: p. 417-494.
2. Shastri, L., *Advances in SHRUTI - a neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony.* Applied Intelligence, 1999. **11**.
3. Wendelken, C. and L. Shastri. *Combining belief and utility in a structured connectionist agent architecture*. in *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society*. 2002.
4. Wendelken, C., *SHRUTI-agent: a structured connectionist architecture for inference and decision-making*, in *Computer Science*. 2003, University of California at Berkeley: Berkeley.
5. Gehring, W.J., et al., *The mind's eye, looking inward? In search of executive control in internal attention shifting.* Psychophysiology, 2003. **40**(4): p. 572-85.
6. Carter, C., et al., *Anterior cingulate cortex, error detection, and the online monitoring of performance.* Science, 1998. **280**: p. 747-749.
7. Goldman-Rakic, P.S., *Cellular basis of working memory.* Neuron, 1995. **14**(3): p. 477-85.
8. Petrides, M., *Frontal lobes and working memory: evidence from investigation of the effects of cortical excisions in nonhuman primates*, in *Handbook of neuropsychology*, F. Boller, H. Spendler, and J. Hendler, Editors. 1994, Elsevier. p. 59-82.
9. Baddeley, A., et al., *Dual-task performance in dysexecutive and nondysexecutive patients with a frontal lesion.* Neuropsychology, 1997. **11**: p. 187-194.
10. Duncan, J. and A. Owen, *Common regions of the human frontal lobe recruited by diverse cognitive demands.* Trends in Neurosciences, 2000. **23**: p. 475-483.
11. Wagner, A.D., et al., *Prefrontal contributions to executive control: fMRI evidence for functional distinctions within lateral Prefrontal cortex.* Neuroimage, 2001. **14**(6): p. 1337-47.
12. Funahashi, S., *Neuronal mechansisms of executive control by the prefrontal cortex.* Neuroscience research, 2001. **39**: p. 147-165.
13. O'Reilly, R., T. Braver, and J. Cohen, *A Biologically Based Computational Model of Working Memory*, in *Models of Working Memory*, A. Miyake and P. Shah, Editors. 1999, Cambridge University Press.
14. Schneider, W., *Working Memory in a Multilevel Hybrid Connectionist Control Architecture*, in *Models of Working Memory*. 1999, Cambridge University Press.
15. Shastri, L., *Advances in SHRUTI -- a neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony.* Applied Intelligence, 1999. **11**.
16. Elias, S. and S. Grossberg, *Pattern formation, contrast control, and oscillations in the short term memory of shunting on-center off-surround networks.* Biological Cybernetics, 1975. **20**: p. 69-98.
17. Wendelken, C. *The role of mid-dorsolateral prefrontal cortex in working memory: a connectionist model*. in *Computational Neuroscience*. 2001.