Institute of Cognitive Science

Technical Report

University of Colorado, Boulder

Making Working Memory Work: A Computational Model of Learning in the Prefrontal Cortex and Basal Ganglia

Randall C. O'Reilly
oreilly@psych.colorado.edu

Department of Psychology
University of Colorado
Boulder, CO 80309

Technical Report 03-03 – Version 12th June, 2003

Abstract

The prefrontal cortex has long been thought to subserve both working memory (the holding of information online for processing) and "executive" functions (deciding how to manipulate working memory and perform processing). Although many computational models of working memory have been developed, the mechanistic basis of executive function remains elusive. In effect, the executive amounts to a homunculus. This paper presents an attempt to deconstruct this homunculus through powerful learning mechanisms that allow a computational model of the prefrontal cortex to control both itself and other brain areas in a strategic, task-appropriate manner. These learning mechanisms are based on structures in the basal ganglia (NAc, VTA, striosomes of the dorsal striatum, SNc) that can modulate learning in other basal ganglia structures (matrisomes of the dorsal striatum, GP, thalamus), which in turn provide a dynamic gating mechanism for controlling prefrontal working memory updating. Computationally, the learning mechanism is designed to simultaneously solve the temporal and structural credit assignment problems. The model's performance compares favorably with standard backpropagation-based temporal learning mechanisms on the challenging 1-2-AX working memory task, and other benchmark working memory tasks.

Introduction

This paper presents a computational model of working memory based on the prefrontal cortex and basal ganglia (the PBWM model). The model represents a convergence of two logically separable but synergistic goals — understanding the complex interactions between the basal ganglia (BG) and prefrontal cortex (PFC) in working memory function, and developing a computationally powerful model of working memory that can learn to perform complex temporally extended tasks. Such tasks require learning which information to maintain over time (and what to forget), and how to assign credit/blame to events based on their temporally delayed consequences. The model shows how the prefrontal cortex and basal ganglia can interact to solve these problems, by implementing a flexible working memory system with a *dynamic gating* mechanism. This mechanism can switch between rapid updating of new information into working memory, and robust maintenance of existing information already being maintained (Hochreiter & Schmidhuber, 1997; O'Reilly, Braver, & Cohen, 1999; Braver & Cohen, 2000; Cohen, Braver, & O'Reilly, 1996; O'Reilly & Munakata, 2000). It is trained in the model using reinforcement learning mechanisms that are widely thought to be supported by the basal ganglia (e.g., Sutton, 1988; Sutton & Barto, 1998; Schultz, Romo, Ljungberg, Mirenowicz, Hollerman, & Dickinson, 1995; Houk, Adams, & Barto, 1995; Schultz, Dayan, & Montague, 1997; Suri, Bargas, & Arbib, 2001; Contreras-Vidal & Schultz, 1999; Joel, Niv, & Ruppin, 2002).

At the biological level of analysis, the PBWM model builds on existing work describing the division of labor between prefrontal cortex and basal ganglia (Frank, Loughry, & O'Reilly, 2001) by adding the critical component of learning. In this prior work, we demonstrated that the basal ganglia can perform dynamic gating via the modulatory mechanism of disinhibition. In the present model, reinforcement learning mechanisms situated in the ventral (limbic) regions of the basal ganglia (specifically the core of the nucleus accumbens, NAc) control the learning of this dynamic gating mechanism. Furthermore, the model shows how the striosome/patch areas contained within dorsal striatum (e.g., Graybiel & Ragsdale, 1978) can provide an additional reinforcement learning signal that helps assign credit/blame to different subsets of working memory representations. In addition to these reinforcement learning mechanisms, the prefrontal cortex representations learn using both Hebbian and error-driven learning mechanisms as incorporated into the Leabra model of cortical learning, which combines a number of well-accepted mechanisms into one coherent framework (O'Reilly, 1998; O'Reilly & Munakata, 2000).

At the computational level, the model is most closely related to the long short term memory (LSTM) model (Hochreiter & Schmidhuber, 1997; Gers, Schmidhuber, & Cummins, 2000), which uses error backpropagation to train dynamic gating signals. The impressive learning ability of the LSTM model compared to other approaches to temporal learning that lack dynamic gating argues for the importance of this kind of mechanism. However, it is somewhat difficult to see how LSTM itself could actually be implemented in the brain. The PBWM model shows how similarly powerful levels of computational learning performance can be achieved using more biologically-based mechanisms.

After presenting the PBWM model and its computational, biological, and cognitive bases, its performance is compared with that of several other standard temporal learning models including LSTM, a simple recurrent network (SRN, Elman, 1990; Jordan, 1986), and real-time recurrent backpropagation learning (RBP, Robinson & Fallside, 1987; Schmidhuber, 1992; Williams & Zipser, 1992).

Working Memory Functional Demands

To contextualize and motivate the model, we can examine a behavioral task called the 1-2-AX task that illustrates three critical functional demands on the working memory system: rapid updating, robust maintenance, and selective updating. Later in the paper, we test the ability of the PBWM model and a variety of other comparison models to learn this task ; further, it has been run in an fMRI experiment on

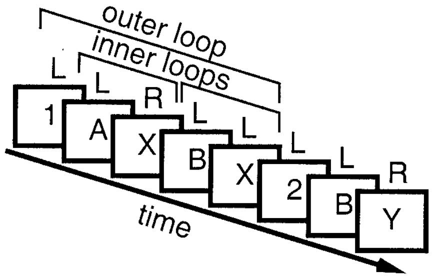


Figure 1: The 1-2-AX task. Stimuli are presented one at a time in a sequence. The participant responds by pressing the right key (R) to the target sequence, otherwise a left key (L) is pressed. If the subject last saw a 1, then the target sequence is an A followed by an X. If a 2 was last seen, then the target is a B followed by a Y. Distractor stimuli (e.g, 3, C, Z) may be presented at any point and are to be ignored. The maintenance of the task stimuli (1 or 2) constitutes a temporal outer-loop around multiple inner-loop memory updates required to detect the target sequence.

human subjects(Kroger, Nystrom, O'Reilly, Noelle, Braver, & Cohen, in preparation). The 1-2-AX task is based on the A-X version of the continuous performance task (AX-CPT), a standard working memory task that has been extensively studied in humans (Servan-Schreiber, Cohen, & Steingard, 1997; Cohen, Barch, Carter, & Servan-Schreiber, 1999; Braver, Barch, Keys, Carter, Cohen, Kaye, Janowsky, Taylor, Yesavage, & Mumenthaler, 2001; Braver & Bongiolatti, 2002; Barch, Braver, Nystrom, Forman, Noll, & Cohen, 1997; Barch, Carter, Braver, Sabb, MacDonald, Noll, & Cohen, 2001; Braver & Cohen, 2001). In AX-CPT, the participant is presented with sequential letter stimuli (A,X,B,Y), and is asked to detect the specific sequence of an A followed by an X by pushing the target (right) button. For all other combinations (A-Y, B-X, B-Y), the participant should respond with a non-target (left) button push. This task requires a relatively simple form of working memory, where the prior stimulus must be maintained over a delay until the next stimulus appears, allowing the participant to discriminate the target from non-target sequences. This is the kind of activation-based working memory that has often been observed in electrophysiological studies of working memory in monkeys (e.g., Fuster & Alexander, 1971; Kubota & Niki, 1971; Miyashita & Chang, 1988; Funahashi, Bruce, & Goldman-Rakic, 1989; Miller, Erickson, & Desimone, 1996).

In the 1-2-AX task (Figure 1), the target sequence varies depending on prior *task demand* stimuli (a 1 or 2). Specifically, if the subject last saw a 1, then the target sequence is A-X. However, if the subject last saw a 2, then the target sequence is B-Y. Thus, the task demand stimuli define an *outer loop* of active maintenance (maintenance of task demands) within which there can be a number of *inner loops* of active maintenance for the A-X level sequences. The three critical functional demands this task imposes on the working memory system are:

Rapid updating: As each stimulus comes in, it must be rapidly encoded in working memory.

Robust maintenance: The task demand stimuli (1 or 2) in the outer loop must be maintained in the face of interference from ongoing processing of inner loop stimuli and irrelevant distractors.

Selective updating: Only some elements of working memory should be updated at any given time, while others are maintained. For example, in the inner loop, A's and X's (etc) should be updated while the task demand stimulus (1 or 2) is maintained.

Dynamic, Selective Gating

The first two functional demands identified above (rapid updating and robust maintenance) are directly in conflict with each other, when viewed in terms of standard neural processing mechanisms (Cohen et al., 1996; Braver & Cohen, 2000; O'Reilly et al., 1999; O'Reilly & Munakata, 2000). Specifically, rapid updating can be achieved by making the connections between stimulus input and working memory strong, but this directly impairs robust maintenance, because such strong connections would allow new stimuli to interfere with ongoing maintenance. Conversely, robust maintenance is best supported by weak input connections relative to the maintenance connections (e.g., recurrent connections among memory units), but this impairs rapid updating of new information. If the

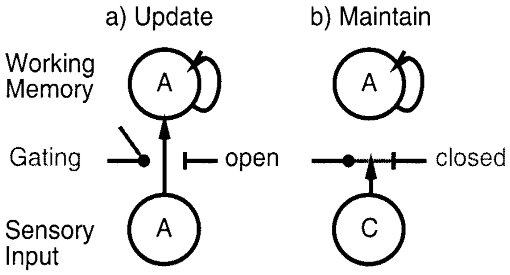


Figure 2: Illustration of active gating. When the gate is open, sensory input can rapidly update working memory (e.g., encoding the cue item A in the 1-2-AX task), but when it is closed, it cannot, thereby preventing other distracting information (e.g., distractor C) from interfering with the maintenance of previously stored information.

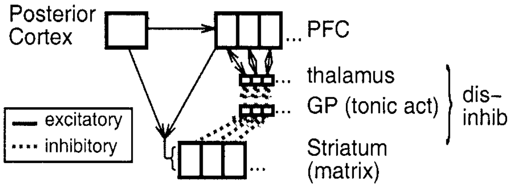


Figure 3: The basal ganglia (striatum, globus pallidus and thalamus) are interconnected with frontal cortex through a series of parallel loops. Striatal neurons disinhibit prefrontal cortex by inhibiting tonically active globus pallidus internal segment (GPi) (and substantia nigra pars reticulata, SNr, not shown) neurons, releasing thalamic neurons from inhibition. This disinhibition provides a modulatory or gating-like function.

inter-unit weights alone determine these connection strengths, and these vary slowly, the model could only learn a suboptimal compromise between these two goals.

A dynamic gating mechanism (Figure 2) avoids these problems by rapidly and flexibly modulating the influence of incoming stimuli on the working memory system (see also Hochreiter & Schmidhuber, 1997). When the gate is open, stimulus information is allowed to flow strongly into the working memory system, thereby achieving rapid updating. When the gate is closed, stimulus information does not strongly influence working memory, thereby allowing robust maintenance in the face of ongoing processing. This gating mechanism must also be selective to allow some information to be robustly maintained (e.g., the outer-loop 1,2 information in the 1-2-AX task), while other information is rapidly updated (e.g., the inner-loop of the 1-2-AX task).

Dynamic Gating of Frontal Maintenance Through Basal Ganglia Disinhibition

One of the central postulates of the PBWM model is that the basal ganglia provide a selective dynamic gating mechanism for information maintained via sustained activation in the PFC (Frank et al., 2001). The hypothesis that the PFC is critical for active maintenance in working memory is almost universally accepted, and is supported by a wide range of cognitive neuroscience data (e.g., Fuster, 1989; Goldman-Rakic, 1987; Miller et al., 1996). The role of the basal ganglia as a gating mechanism that modulates this prefrontal active maintenance system is also consistent with a considerable amount of biological and behavioral data, as reviewed in Frank et al. (2001). A few key examples are summarized here.

First, anatomically (with some simplification; see Frank et al., 2001 for details), the *direct pathway* through the dorsal striatum, globus pallidus (GP), thalamus, and back to PFC provides a disinhibitory modulation of PFC (Figure 3). GP neurons are tonically active and thus tonically inhibit the thalamus. When a striatal neuron fires (they are usually inactive), it inhibits the GP neurons to which it projects, thus disinhibiting the thalamus, which is reciprocally interconnected with the PFC via excitatory connections. This thalamic disinhibition thus enables, but does not directly cause (i.e., gates), a loop of excitation into the PFC. The effect of this excitation in the model is to toggle the state of bistable currents in the PFC neurons. Thus, when PFC neurons are in the *up* state, they have a persistent excitatory current that helps them remain active over time, while other neurons in the *down* state lack this extra excitation (Fellous, Wang, & Lisman, 1998; Wang, 1999; Durstewitz, Kelc, & Gunturkun, 1999; Durstewitz, Seamans, & Sejnowski, 2000a). This intracellular maintenance is further supported by recurrent excitatory connections among PFC neurons, and the combination provides important computational advantages (Frank et al., 2001).

In short, the firing of a direct-pathway neuron, which we refer to as a GO signal (and the neurons as GO neurons), toggles the maintenance of information in PFC. To clear an existing representation and store a different one (i.e., an update), two GO signals are required. This toggling pattern of behavior has been observed in PFC neurons *in vitro* (J. Seamans, personal communication, January 2002). There are also striatal neurons that project via an *indirect pathway*, with the effect of increasing the level of inhibition on the thalamic pathway. We refer to these as the *NO-GO* neurons in the model — they compete with the GO neurons and enable the PFC to continue to maintain currently stored information. This competition, and competition between different possible GO signals across different basal ganglia areas, is likely mediated within the GP and subthalamic nucleus circuitry, not directly in striatum as has been otherwise proposed (e.g., Mink, 1996; Wickens, 1993).

Critically, the basal ganglia can provide a *selective* gating mechanism because there are parallel loops of connectivity through different areas of the basal ganglia and frontal cortex (Alexander, DeLong, & Strick, 1986; Graybiel & Kimura, 1995; Middleton & Strick, 2000). Thus, different regions of PFC can be updated independently by different regions of the basal ganglia — in the context of the 1-2-AX task, this would predict that different regions of PFC are used to hold the outer loop information, which needs to be maintained while the inner loop is updated. Indeed, fMRI evidence supports this prediction in this task (Kroger et al., in preparation), and in other tasks with an inner/outer loop structure (e.g., Braver & Bongiolatti, 2002; Koechlin, Corrado, & Grafman, 2000). We refer to the separately updatable components of the PFC/BG system as *stripes*, in reference to relatively isolated groups of interconnected neurons in PFC (Levitt, Lewis, Yoshioka, & Lund, 1993; Pucak, Levitt, Lund, & Lewis, 1996). We previously estimated that the human frontal cortex could support roughly 20,000 such stripes (Frank et al., 2001).

Summary and Application to the 1-2-AX Task

Figure 4 shows how the BG-mediated selective gating mechanism can enable performance of the 1-2-AX task (see Frank et al., 2001 for a working simulation). When a task demand stimulus is presented (e.g., 1), a BG gating signal (i.e., a GO signal) must be activated to enable a particular PFC stripe to retain this information (panel a). A *different* stripe must be gated for the subsequent cue stimulus A (panel c), and no stripe (or NO-GO firing) should be activated for a distractor such as C (panel b). When the X stimulus is presented, the combination of this stimulus representation plus the maintained PFC working memory representations is sufficient to trigger a target response R (panel d). Note that this motor response is triggered using the same disinhibitory gating mechanism as was involved in working memory gating; it affects more posterior frontal areas (e.g., SMA) that drive responding.

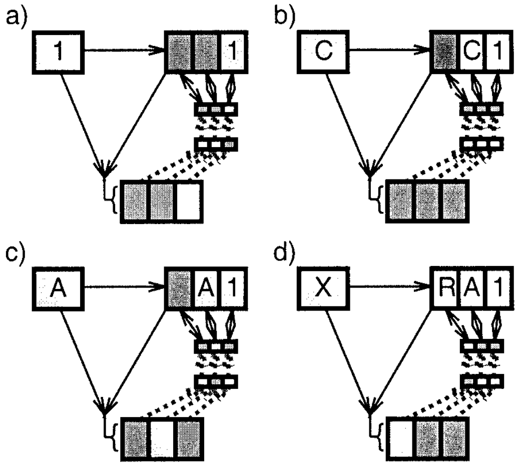


Figure 4: Illustration of how the basal ganglia gating of different PFC stripes can solve the 1-2-AX task (light color = active; dark = not-active). **a)** The 1 task is gated into an anterior PFC stripe because a corresponding striatal stripe fired. **b)** The distractor C fails to activate the BG so it will not be maintained; however, it does elicit transient PFC activity. Note that the 1 persists because of gating-induced robust maintenance. **c)** The A is gated in. **d)** A right keypress motor action is activated (using same BG-mediated disinhibition mechanism) based on X input plus maintained PFC context.

Learning When to Gate in the Basal Ganglia

As Figure 4 makes clear, the learning problem in the basal ganglia amounts to learning when to fire a GO vs. NO-GO signal in a given stripe based on the current sensory input and maintained PFC activations. Without such a learning mechanism, which can develop from initially random gating behavior into a strategic, task-appropriate pattern of gating, our model would require some kind of intelligent homunculus to control gating. Thus, the development of this learning mechanism is a key step in banishing the homunculus from the domain of working memory models (c.f., the "central executive" of Baddeley's (1986) model). There are two fundamental problems that must be solved by the learning mechanism:

Temporal credit assignment: The benefits of having encoded a given piece of information into prefrontal working memory are typically only available later in time (e.g., encoding the 1 task demand only helps later when confronted with an A-X sequence). Thus, the problem is to know which prior events were critical for subsequent good (or bad) performance.

Structural credit assignment: The network must decide which stripes should encode which different pieces of information, and when successful performance occurs, it must reinforce those stripes that actually contributed to this success. This form of credit assignment is what neural network models are typically very good at doing, but clearly this form of structural credit assignment interacts with the temporal credit assignment problem, making it more complex.

The solutions to these problems adopted in the PBWM model (illustrated abstractly in Figure 5 and in biological detail in Figure 6) are inspired by two important aspects of the basal ganglia biology. First, we adopt the temporal differences (TD) reinforcement learning mechanism (Sutton, 1988; Sutton & Barto, 1998) as a model of the ventral striatum (nucleus accumbens; NAc) and its control over the firing of ventral tegmental (VTA) and substantia nigra pars compacta (SNc) dopamine (DA) neurons.

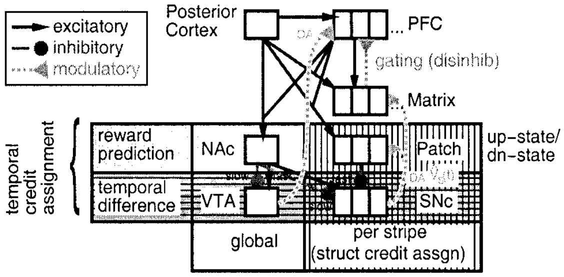


Figure 5: Solution of the temporal credit assignment and structural credit assignment problems via different parts of the basal ganglia system. The nucleus accumbens (NAc) and ventral tegmental area (VTA) perform global temporal differences learning by learning to expect rewards and computing temporal derivatives, respectively. The patch regions of the striatum (striosomes) learn the expected reward values on a per-stripe basis ($\hat{V}_s(t)$), and modulate the temporal difference computation in the substantia nigra pars compacta (SNc) (which is based on inputs from the NAc). The result of this modulation is that negative VTA TD values are moderated when expected reward is high (i.e., when the network has learned to have high confidence in a given stripe, other stripes are blamed for errors).

The TD mechanism is designed to solve the temporal credit assignment problem, and it is widely thought to explain aspects of the firing properties of the VTA DA neurons (e.g., Schultz et al., 1995; Houk et al., 1995; Schultz et al., 1997; Suri et al., 2001; Contreras-Vidal & Schultz, 1999; Joel et al., 2002).

The basic TD mechanism is sufficient to drive competent learning in the PBWM model, at least in some cases (see results below). However, the basal ganglia has considerable additional circuitry involving the striosome (patch) areas of the dorsal striatum (Figure 6) that is anatomically capable of modulating the overall TD signal computed by the NAc (e.g., Graybiel & Ragsdale, 1978). Computationally, it would make sense if these areas were to provide a stripe-specific modulation of the global NAc signal, to help solve the structural credit assignment problem. After exploring a range of possible such stripe-specific modulations, we found one (and only one) that appears to provide reliable computational benefits. This stripe-specific signal is computed from the expected reward value for a given stripe only when that stripe has been actively main-

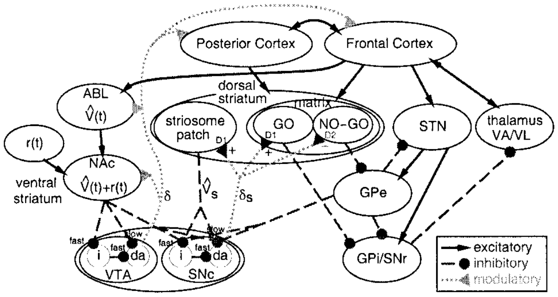


Figure 6: Detailed basal ganglia/frontal cortex circuitry represented in the model, with hypothesized computational quantities (defined in the text) associated with different projections. The dorsal striatum contains both matrix and patch/striosome components, and the matrix is further subdivided into GO and NO-GO units, which project direct disinhibition or indirect inhibition onto the thalamus, respectively. The STN (subthalamic nucleus) provides an additional dynamic background of inhibition (NO-GO) by exciting the GPi/SNr (globus pallidus internal segment/substantia nigra pars reticulata). The ventral striatum nucleus accumbens (NAc) drives the VTA dopamine system and the SNc (substantia nigra pars compacta); SNc is also modulated by the striosomes, producing a stripe-wise dopamine signal back to the dorsal striatum. Raw expected reward values ($\hat{V}(t)$) may be computed in the ABL (basolateral nucleus of the amygdala), while the NAc integrates these expectations with actual reward outcomes. The derivative of NAc states can be computed by fast GABA-A mediated disinhibition (via interneurons i) followed by slower direct GABA-B mediated inhibition.

taining information in working memory. When a negative global TD signal (i.e., an error) occurs, a strong stripe-specific expected reward value protects a given stripe from this "blame" signal. In effect, when an error occurs, the striosome/SNc system says "blame the other stripes" because this stripe has been very effective in producing correct responses in the past.

This division of labor between the NAc and the striosomes is distinct from existing TD models of the basal ganglia, which have typically focused only on the striosomes as the source of TD computations. As emphasized by Joel et al. (2002), the neural connections from the striosomes to the SNc may not support the temporal derivative computation necessary for the TD algorithm. Our modulatory proposal for the striosome/SNc system might provide a resolution to this problem. Further, we suggest that the necessary temporal derivative computation in the VTA and SNc can be performed via a combination of direct inhibitory projections and indirect inhibitory projections to inhibitory interneurons within the VTA (Charara, Heilman, Levey, & Smith, 1999). Other mechanisms for this computation are also possible, including projections via the ventral pallidum (Pennartz, Groenewegen, & Lopes da Silva, 1994). The NAc provides the primary source of input to the VTA, and it has been strongly implicated in stimulus-response motor learning of the form supported by the TD algorithm (e.g., Hernandez, Sadeghian, & Kelley, 2002; Kelley, Smith-Roe, & Holahan, 1997).

Learning Mechanism Details

The fundamental computation of the TD algorithm is to predict subsequent rewards based on current sensory inputs — it learns to find the earliest reliable predictors of subsequent reward. Specifically, TD has the desired effect of moving the reward-driven learning signal from the point where reward is actually delivered to the earliest point where the reward can be predicted. In the 1-2-AX task, for example, it can learn to apply the subsequent success associated with having encoded the 1 task demand stimulus to the point in time when the 1 actually appears, thereby reinforcing the firing of an appropriate GO unit in the matrix of the dorsal striatum to ensure storage on later trials.

TD is defined in terms of a value function $V(t)$ that represents the sum of all future rewards r at a given state indexed by a point in time t:

$$V(t) = \sum_{\tau=t+1}^{\tau=\infty} \gamma^{\tau-(t+1)} r(\tau) \qquad (1)$$

where $0 \leq \gamma \leq 1$ is a discounting factor that values more distant future rewards less than more proximal future rewards, and causes the sum to converge.

Of course, the organism never knows the $V(t)$ values directly, and instead must learn to estimate them — this is what the TD algorithm does. It can be derived by writing the value function sum recursively:

$$V(t) = r(t+1) + \gamma V(t+1) \qquad (2)$$

and then using this recursive definition to update estimates of the values associated with each state, $\hat{V}(t)$. Specifically, TD trains these estimates by making the two sides of equation 2 consistent, which is to say, by minimizing their difference:

$$\delta(t) = [r(t+1) + \gamma\hat{V}(t+1)] - \hat{V}(t) \quad (3)$$

Computationally, we cannot know the future, so we really run this computation at time t with reference to the *previous* point in time:

$$\delta(t-1) = [r(t) + \gamma\hat{V}(t)] - \hat{V}(t-1) \quad (4)$$

The $\delta(t-1)$ value is the temporal differences (TD) error between reward information at times t and $t-1$, and minimizing this difference (by moving $\hat{V}(t-1)$ in the direction of $\delta(t-1)$) results in better value function estimates $\hat{V}(t)$.

The TD error is typically used in an *actor-critic* framework (Sutton & Barto, 1998). The *critic* in this framework is the aforementioned system that computes the value function estimates and their temporal derivatives, and corresponds to the NAc/VTA system in the PBWM model. The $\delta(t-1)$ value computed by the critic can also be used to modulate the learning of *actor* units that produce the actions that lead to rewards (i.e., the GO/NO-GO units in the matrix of the dorsal striatum in our case). A simple and widely used such learning rule is:

$$\Delta w_{ij}(t-1) = \delta(t-1)y_j(t)x_i(t-1) \quad (5)$$

where $x_i(t-1)$ is the activation of a sending unit connected with weight w_{ij} to receiving unit with activation $y_j(t)$. Thus, if the TD error is positive (i.e., more reward is expected/obtained now than was previously expected), the weights from active sending units are increased, and if less reward is expected/obtained now than was previously expected, weights from active sending units are decreased. This is the learning rule employed in the basal ganglia units of the PBWM model.

To summarize the global TD computation in the model, the NAc activations reflect expectations of future rewards and experiences of actual rewards, while the VTA computes the temporal derivative of the NAc states ($\delta(t-1)$, equation 4). The VTA dopamine neurons then project back to the NAc to train its reward estimates (and to the cortex). The striosomes modulate the global TD signal in corresponding SNc neurons based on stripe-wise expected reward values, and the DA signals from the SNc train the working memory update signals computed by the matrisomes. This general arrangement is in good agreement with the connectivity between these brain areas (Figure 6).

Structural Credit Assignment via the Striosome/SNc System

The global TD mechanism described above computes a single scalar value, $\delta(t-1)$, that is then applied uniformly to all of the actor units in the network. However, not all actor units in the system are equally culpable for errors that are made. This is especially true when multiple actor systems are working in parallel, as in the PBWM model where each stripe can be maintaining different pieces of information relevant to the overall task. For example, one stripe might be correctly maintaining the 1 task demand information, while another stripe fails to maintain the A stimulus and thus produces an error. In this case, it would make sense to punish the *A* stripe and not the 1 stripe. This is exactly what we propose the striosome/SNc system achieves[1].

Specifically, we assume that there are distinct sets of striosome (patch) and matrisome (matrix) neurons in the dorsal striatum associated with each separately updatable stripe. The striosome units compute expected rewards just like the NAc units, except that their activation is modulated by the extent to which the GO neurons in the corresponding matrix region have recently fired. When a GO signal is computed by the matrix, this puts neurons in the corresponding striosome into an up-state (e.g., Plenz & Kitai, 1998; Stern, Kincaid, & Wilson, 1997; Cowan & Wilson, 1994), enabling them to become active and to learn reward expectations. This up-state transition signal might be mediated by dopamine inputs from the SNc (Surmeier & Kitai, 1999), or the large cholinergic interneurons within the dorsal striatum. The striosome neurons

[1] This general idea of protecting representations from error was originally suggested to the author by Clay Holroyd, personal communication, 2001.

learn to produce good estimates of expected reward, conditional on when their corresponding stripe is participating in maintaining information (i.e., when they are in the up-state). The striosome projections to the SNc directly convey these stripe-wise expected reward values, $\hat{V}_s(t-1)$, which combine with the global TD $\delta(t-1)$ value computed by the NAc (which drives the SNc neurons directly via the same projections that drive the VTA; Pennartz et al., 1994):

$$\delta_s(t-1) = \begin{cases} \delta(t-1) & \delta(t-1) > 0 \\ \delta(t-1)(1-[\hat{V}_s(t-1)]_+) & \delta(t-1) \leq 0 \end{cases} \tag{6}$$

where $[x]_+ = x$ if $x > 0$ and 0 otherwise. In short, a positive $\hat{V}_s(t-1)$ value will diminish the magnitude of a negative $\delta(t-1)$ TD value. The $\delta_s(t)$ value is used to train both the striosomes and matrisomes of the corresponding stripe, replacing $\delta(t-1)$ in equation 5.

Although this formulation of the role of the striosome/SNc system was generally inspired by the anatomical connectivity of these areas, the details of equation 6 represent a prediction going beyond available data. As demonstrated later, this role for the striosome/SNc system provides some computational benefits, whereas a number of other alternative ideas that were explored did not. Anatomically, this mechanism predicts stripe-selective dopamine projections from the SNc to the dorsal striatum. Physiologically, it predicts differential firing patterns from different SNc neurons associated with different stripes, but only for negative global TD signals (i.e., corresponding to a decrease from baseline of the tonically firing dopamine neurons). The protective effect of striosomal projections into SNc could be realized via a shunting effect on the SNc dopamine neurons. For example, if striosomes provided strong inhibitory inputs to both the SNc DA neurons and their inhibitory interneurons, these offsetting currents would shunt or dilute any TD inputs from the NAc, without producing a net excitatory or inhibitory drive. Although clearly speculative at this point, these biological properties stand as testable predictions of the PBWM model.

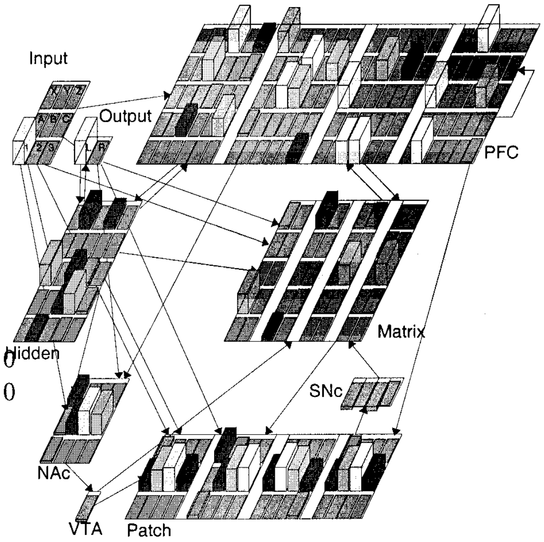


Figure 7: Implemented model as applied to the 1-2-AX task. There are 4 stripes in this model as indicated by the groups of units within the PFC, Patch, Matrix, and SNc. The left-hand column of each matrix stripe represents GO units, while the right hand are the NO-GO units. The ImRew, NAc, and Patch units represent scalar reward prediction values using distributed coarse-coded representations, while the dRew, and VTA compute the temporal derivatives of the ImRew and NAc layers, respectively.

Details of the PBWM Implementation

The model, shown in Figure 7, is implemented using the Leabra framework, described in detail in the Appendix (O'Reilly, 1998; O'Reilly & Munakata, 2000; O'Reilly, 2001). Leabra uses point neurons with excitatory, inhibitory, and leak conductances contributing to an integrated membrane potential, which is then thresholded and transformed via an $x/(x+1)$ sigmoidal function to produce a rate code output communicated to other units (discrete spiking can also be used, but produces noisier results). Each layer uses a k-winners-take-all (kWTA) function that computes an inhibitory conductance that keeps roughly the k most active units above firing threshold and keeps the rest below threshold. Units learn according to a combination of Hebbian and error-driven learning, with the latter computed using the generalized recirculation algorithm (GeneRec; O'Reilly, 1996), which computes backpropagation derivatives using two phases of activation settling as in the deterministic Boltzmann machine and contrastive Hebbian learning algorithms (Hinton, 1989; Movellan, 1990). The *minus* phase represents the network's expectation or response in the current situation, and the *plus* phase represents a subsequent outcome or result. In the 1-2-AX task, the minus phase is just the network's output response (L or R) and the plus phase is the correct response. The cortical layers in the model use standard Leabra parameters and functionality, while the basal ganglia systems require some additional mechanisms, detailed next.

The reward prediction layers (NAc and Patch) use a distributed, coarse-coded representation of the scalar values they encode ($\hat{V}$ or $\hat{V}_s$). Thus, each unit has a preferred value with a graded Gaussian tuning curve around it (except the first unit, which reflects the decoded scalar value for display purposes, but does not otherwise participate in the network computation). This way of representing scalar values, instead of the typical use of a single unit with linear activations, allows much more complex mappings to be learned. For example, units representing high values can have completely different patterns of weights than those encoding low values, whereas a single unit is constrained by virtue of having one set of weights to have a monotonic mapping onto scalar values. Although this limitation could be remedied by having a hidden layer prior to each scalar value unit, the coarse-coded representations simplify the network architecture and are biologically plausible in any case.

The computation of the TD algorithm takes place over the sequence of minus and plus phases of the GeneRec algorithm, as follows. The NAc units are clamped in the minus phase to the prior time step's expected reward value ($\hat{V}(t-1)$), but are free to settle in the plus phase to compute the expected reward value for this time step ($\hat{V}(t)$). This value is saved and is used for clamping the minus phase at the next time step (i.e., it becomes $\hat{V}(t-1)$ at time $t+1$). At the end of the plus phase, the NAc value is multiplied by the discounting factor γ and the actual reward value ($r(t)$) added. Thus, the plus-minus phase difference in activations within the NAc is the TD delta $\delta(t-1)$ (equation 4), and this is what drives learning of the NAc units (using sending activations from the prior time step, as dictated by equation 5). The VTA and SNc units directly take the temporal minus-plus phase difference to compute the $\delta(t-1)$ value. The global VTA $\delta(t-1)$ value can be used to modulate cortical learning, as likely occurs in the brain. Although this modulation was not directly implemented for the simulations reported here, it is likely that this VTA dopamine signal plays an important role in the brain achieving something like the basic GeneRec error-driven learning mechanisms used in the PBWM model (O'Reilly, 1996; O'Reilly & Munakata, 2000).

The SNc $\delta(t-1)_s$ values (equation 6) are modulated by inputs from the patch units, which are trained to encode $\hat{V}_s(t)$ values through an interaction between matrix GO-unit firing and $\delta(t-1)_s$ values from the SNc (thus, the patch is self-regulating). Specifically, when a GO unit fires (toggling PFC maintenance on), it puts the striosomes into an up-state whereby they can compute expected rewards much like the NAc units (the minus phase value is $\hat{V}_s(t-1)$, and the plus phase is the settled value plus $\delta(t-1)_s$. The up-state lasts until another GO unit fires (toggling PFC maintenance off). The protected TD error values $\delta_s(t-1)$ computed by the SNc units are sent to the matrix (and patch) units in the corresponding stripe. The matrix units use this $\delta_s(t-1)$ value to drive learning on their incoming connections (again using equation 5) — this is the ultimate role of the entire set of critic units described to this point.

The direct and indirect pathways that mediate the GO/NO-GO gating effects of the matrix units are abstracted via a function that directly toggles PFC intracellular maintenance currents (i.e., up/down states) in response to GO-unit firing. The GO and NO-GO units compete in the matrix via a strong kWTA competition within each stripe, such that generally only one GO or NO-GO unit can be active. Thus if any GO unit within a stripe fires, it directly toggles an intracellular excitatory ionic conductance on or off within each of the currently active PFC units. This excitatory ionic conductance persists until the next GO firing, and provides a bias (along with the recurrent excitatory self-connections among PFC units) for those units to remain active (see Frank et al., 2001 for further discussion of this kind of maintenance mechanism, which has been proposed by several researchers e.g., Lewis & O'Donnell, 2000; Fellous et al., 1998; Wang, 1999; Dilmore, Gutkin, & Ermentrout, 1999; Gorelova & Yang, 2000; Durstewitz, Seamans, & Sejnowski, 2000b). The only effect of NO-GO firing is to prevent GO unit firing.

To achieve a complete update of the PFC units in one event (stimulus) presentation, a third phase that consists of updating PFC representations must occur after the standard minus-plus sequence. All other units in the network remain unchanged during this phase. This phase-wise structuring represents a discretization of a more continuous process in the brain, where PFC representation updates lag those of posterior cortex. Specifically, during the minus and plus phase, the PFC units settle like any other units in the cortical system, and any maintenance currents remain as they were. The PFC thus provides a stable context memory of prior information during the processing of the current event. After the plus phase, if a matrix GO unit has fired, then this toggles maintenance in the PFC to the opposite state. The additional phase of settling allows PFC units that were toggled off to settle into a new activation state that reflects the current input. If at the end of this phase a matrix GO unit again fires, then this new activation pattern will be maintained through maintenance currents in the activated units. However, if a NO-GO unit fires, the PFC representations will not be maintained, and will simply reflect incoming sensory inputs until maintenance is again activated on a later trial.

Application: The 1-2-AX Task

The PBWM model (Figure 7) and several comparison networks (various flavors of recurrent backpropagation) were trained on the 1-2-AX task, to evaluate how well this biologically-based mechanism performs relative to simpler but biologically implausible learning mechanisms. In addition, variants of the PBWM network were run to explore the role of the different features of the network.

The task was trained as in Figure 1, with the length of the inner loop sequences randomly varied from one to four (i.e., one to four pairs of A-X, B-Y, etc stimuli). Specifically, each sequence of stimuli was generated by first randomly picking a 1 or 2, and then looping for one to four times over the following inner-loop generation routine. Half of the time (randomly selected), a possible target sequence (either A-X or B-Y) was generated. The other half of the time, a random sequence composed of an A, B, or C followed by an X, Y, or Z was randomly generated. Thus, possible targets (A-X, B-Y) represent at least 50% of trials, but actual targets (A-X in the 1 task, B-Y in the 2 task) appear only 25% of time on average. The correct output was the L unit except on the target sequences (1-A-X or 2-B-Y), where it was an R. The PBWM network received a reward if it produced the correct output (and received the correct output on the output layer in the plus phase of each trial), while the backpropagation networks learned from the error signal computed relative to this correct output. One epoch of training consisted of 25 outer-loop sequences, and the training criterion was 0 errors across one epoch. Training was stopped after 1,000 epochs for the PBWM models and 10,000 epochs for the backpropagation models if the network had failed to learn by this point, and was scored as a failure to learn (PBWM takes more computer time per epoch of training, and typically learns within 1,000 epochs or not at all).

Comparison with Backpropagation-Based Networks

The networks compared were:

- The full PBWM model with 8 stripes (30 PFC units and 10 matrix units per stripe) and 49 hidden units.
- A simple recurrent network (SRN, Elman, 1990; Jordan, 1986) with 100 hidden units and 100 context units, cross-entropy output error, learning rate of .1 (no momentum), an error tolerance of .1 (output err $<$ .1 counts as 0), and a hysteresis term in updating the context layers of .5 ($c_j(t) = .5h_j(t-1) + .5c_j(t-1)$, where c_j is the context unit for hidden unit activation h_j). Learning rate, hysteresis, and hidden unit size were searched for

optimal values across this and the RBP networks (within plausible ranges, using round numbers, e.g., lrates of .05, .1, .2, and .5; hysteresis of 0, .1, .2, .3, .5, and .7, hidden units of 25, 36, 49, and 100). Optimal performance was with 100 hidden units, hysteresis of .5, and lrate of .1.

- A real-time recurrent backpropagation learning network (RBP, Robinson & Fallside, 1987; Schmidhuber, 1992; Williams & Zipser, 1992), with the same basic parameters as the SRN, and a time constant for integrating activations and backpropagated errors of 1, and the gap between backpropagations and the bačkprop time window searched in the set of 6, 8, 10, and 16 time steps. Two time steps were required for activation to propagate from the input to the output, so the effective backpropagation time window across discrete input events in the sequence is half of the actual time window (e.g., 16 = 8 events, which represents 2 or more outer-loop sequences). Best performance was achieved with the longest time window (16).
- A long short term memory (LSTM) model (Hochreiter & Schmidhuber, 1997) with forget gates as specified in Gers et al. (2000), with the same basic backpropagation parameters as the other networks, and 4 memory cells.

The basic results for number of epochs required to reach a criterion training level of 0 errors across one epoch of 25 outer-loop sequences are shown in Figure 8. These results show that the PBWM model learns the task somewhat faster than the comparison backpropagation networks. However, the main point is not in comparing the quantitative rates of learning (it is possible that other parameters could be found to make the comparison networks perform better). Rather, these results simply demonstrate that the biologically-based PBWM model is in the same league as existing powerful computational learning mechanisms.

Furthermore, the exploration of parameters for the backpropagation networks demonstrate that

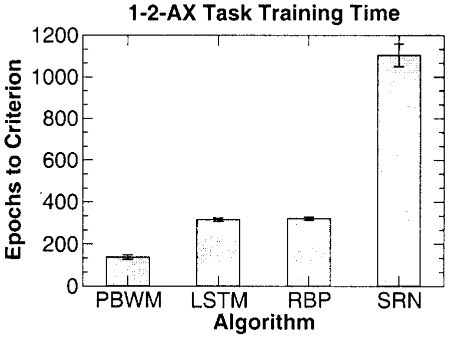


Figure 8: Training time to reach criterion (0 errors in one epoch of 25 outer-loop sequences) on the 1-2-AX task for the PBWM model and three backpropagation-based comparison algorithms. LSTM = long short-term memory model, RBP = recurrent backpropagation (real time recurrent learning), SRN = simple recurrent network.

a) Hidden layer sizes for SRN (lrate = .1, hyst = .5):

hiddens:	25	36	49	100
Failures	60%	4%	0%	0%
Avg Epochs	4,228	2,849	1,926	1,104

b) Hysteresis for SRN (100 hiddens, lrate = .1):

hyst:	.1	.2	.3	.5	.7
Failures	100%	66%	0%	0%	0%
Avg Epochs	n/a	5,135	2,207	1,104	1,187

c) Learning rates for SRN (100 hiddens, hyst = .5):

lrate:	.05	.1	.2
Failures	0%	0%	0%
Avg Epochs	1,380	1,104	1,231

Table 1: Effects of various parameters on learning performance in the SRN. Failures is number of networks (out of 50) that failed to learn to criterion (0 errors for an epoch) within 10,000 epochs, and Avg Epochs is average number of epochs to reach criterion for successful networks. The optimal performance is with 100 hidden units, learning rate .1, and hysteresis .5. Sufficiently large values for the hidden units and hysteresis parameters are critical for successful learning, indicating the strong working memory demands of this task.

a) Time window for RBP (lrate = .1, 100 hiddens):

window:	6	8	10	16
Failures	68%	2%	0%	0%
Avg Epochs	1,311	503	384	322

b) Hidden layer size for RBP (lrate = .1, window = 16):

hiddens:	25	36	49	100
Failures	0%	0%	0%	0%
Avg Epochs	720	592	428	322

Table 2: Effects of various parameters on learning performance in the RBP network. The optimal performance is with 100 hidden units, time window = 16. As with the SRN, the relatively large size of the network and long time windows required indicate the strong working memory demands of the task.

the 1-2-AX task represents a challenging working memory task, requiring large numbers of hidden units and long temporal-integration parameters for successful learning. For example, the SRN network started to show learning failures (within 10,000 epochs) with 36 hidden units or less, and with hysteresis parameters (which determines the window of temporal integration of the context units) below .3 (Table 1). Optimal parameters for the SRN appeared to be 100 hidden units, hysteresis of .5, and a learning rate of .1. For the RBP network, the number of hidden units and the time window for backpropagation exhibited similar results (Table 2). Specifically, time windows of eight or fewer time steps resulted in failures to learn, and best results were achieved with the most hidden units and the longest backpropagation time window.

Testing the Patch/SNc System in the PBWM Model

The contribution of the striosome (patch)/SNc error-protection mechanism in the PBWM model was tested by comparing models lacking this feature to the full model. The results (Figure 9, Interleaved Training) suggest that this mechanism plays some role, but it was not particularly dramatic, amounting to a single failed network out of 50 (2%). Overall differences in number of epochs to criterion, even when including the 1,000 epoch score for the failed network, were not statistically significant, although the patch/SNc model did have a lower overall mean training time (138.4 vs. 157.98).

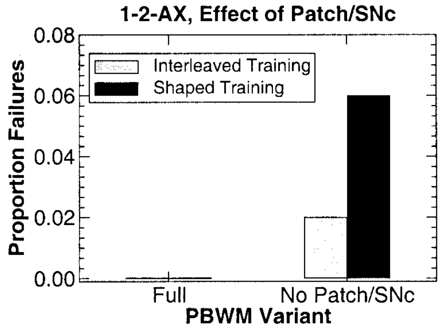


Figure 9: Proportion of networks failing to learn to 0 error criterion (out of 50 runs), for the full PBWM network as compared to a version without the Patch/SNc system. The Interleaved Training results are for the 1-2-AX task as normally trained, while the Shaped Training results are the "shaped" version of the 1-2-AX task where the network is initially trained to respond to the X, then A followed by X, then only A-X when preceded by a 1, etc.

To provide a stronger test of the value of the error-protection mechanism, networks were trained using a "shaping" schedule of piece-wise introduction of task elements, instead of presenting the full task at the beginning. The error-protection mechanism was predicted to be especially important with this schedule, for preserving the already-learned aspects of the task as new task elements are introduced. This proved to be the case.

Specifically, we trained the networks in five stages, designed so that stimuli that would later become task-relevant were not first introduced in a task-irrelevant fashion. First, networks were trained to respond R for X and L for non-X (only X and Z were presented). Then, the R target was only X when preceded by an A (only A-X, A-Z, C-X, C-Z sequences were presented). Then R responses were expanded to also include Y targets. Next, R was either an A-X or B-Y sequence. Finally, the full 1-2-AX task was trained. Importantly, as each new element of the task is introduced, the network needs to retain the successful actions from prior learning, and apply the error and reward feedback to shaping new actions. This is exactly what the striosome/SNc mechanism is designed to do, and the results (Figure 9, Shaped Training) show that indeed this increased the failure rate of the model lacking the striosome/SNc mechanism by more than a factor of two.

It may appear ironic that using the shaping actually impairs performance, but this is consistent with a wide range of computational modeling results suggesting that learning of tasks is better when all elements are interleaved from the beginning (e.g., McClelland, McNaughton, & O'Reilly, 1995). In the real animal, shaping is critical for motivational purposes, and such motivational factors have yet to be included in the present model. Therefore, the extra benefits of the striosome/SNc mechanism in the shaping context may be particularly important as such elements are also introduced into the model, and training is performed in a highly incremental, temporally extended manner (as in human development).

Application: The SIR Task

The PBWM and comparison backpropagation algorithms were also tested on a more commonly-used type of task for testing the ability of a dynamic gating function to maintain information over long time delays. In this task, called the store ignore recall (SIR) task, the network must store an arbitrary input pattern for a recall test that occurs after a variable number of intervening ignore trials. Stimuli are presented during the ignore trials, and must be identified (output) by the network, but do not need to be maintained. Tasks with this same basic structure were the focus of the original Hochreiter and Schmidhuber (1997) work on the LSTM algorithm, where they demonstrated that the dynamic gating mechanism was able to gate in the to-be-stored stimulus, maintain it in the face of an essentially arbitrary number of intervening trials by having the gate turned off, and then recall the maintained stimulus. The SIR version of this task can be considered a paradigmatic example of a working memory task (O'Reilly & Munakata, 2000).

The Hochreiter and Schmidhuber (1997) version of this basic task may have provided a bit of a crutch for the gating network, in that the critical item to be stored was always the first in a sequence, and the network activations were initialized after each sequence. Thus, the network only needed to learn to open the gate for the first item in the sequence, and to keep the gate off at all other times. Given the highly distinctive differences in activation patterns present at the start of the sequence (i.e., everything off) compared to subsequent trials, this may have been relatively easy to learn. In the present version of this task, by contrast, stimuli are presented in a continuous stream with no activation initialization, and input units are activated to indicate when to store, ignore, and recall. Thus, the network has to learn the significance of these inputs (which are otherwise identical to the other stimulus inputs) in order to solve the task. In summary, the networks had 7 input units, 3 of which were the control inputs (S,I,R) and the remaining 4 were stimulus items (A-D). The durations between store and recall trials (i.e., maintenance durations) were randomly sampled from a uniform distribution between 1 and a maximum value of 4, 8, or 16. The number of ignore trials between recall and store trials was randomly chosen between 0 and 2. A typical sequence of inputs and target outputs is shown in Figure 10.

Trial	Input	Maint	Output
1	I-D	–	D
2	S-A	A	A
3	I-B	A	B
4	I-C	A	C
5	I-D	A	D
6	R	A	A
7	I-A	–	A
8	I-C	–	C
9	S-D	D	D

Figure 10: An example sequence of trials in the SIR task, showing what is input, what should be maintained, and the target output. I = Ignore unit active, S = Store unit active, R = Recall unit active. The functional meaning of these "task control" inputs must be discovered by the network, and differentiated from the otherwise identical A-D stimulus inputs, through learning.

As Figure 11 indicates, the two algorithms with dynamic gating mechanisms (the PBWM and LSTM models) are only slightly affected by increases in the maximum maintenance duration between store and recall trials, while the other net-

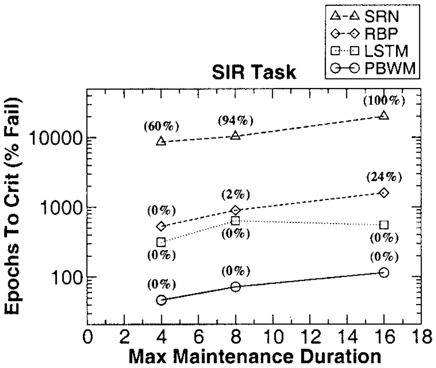
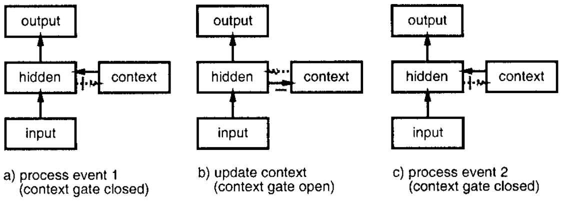


Figure 11: Results (epochs to criterion and, in parentheses above each data point, percent failures to learn within 20,000 epochs) for all algorithms on the SIR (store ignore recall) task, as a function of maximum maintenance duration between storage and recall. Note the logarithmic plot of the Y axis. The algorithms with dynamic gating (PBWM, LSTM) are only slightly affected by maintenance duration, while the other networks suffer dramatically.

works lacking this mechanism start to fail as the maintenance duration increases. This replicates the basic findings of Hochreiter and Schmidhuber (1997), and reinforces the importance of dynamic gating mechanisms for robust working memory maintenance. Furthermore, it generalizes the results from the 1-2-AX task on a more standard benchmark task, again showing that the PBWM algorithm is capable of rapid learning of working memory maintenance strategies. This task also showed some very modest benefits for the striosome/SNc system in the PBWM model, with the model lacking this mechanism exhibiting a 2% failure rate on the 8 max maintenance duration problem.

Discussion

The PBWM model presented here demonstrates powerful learning abilities on demonstrably complex and difficult working memory tasks. We have also tested it informally on a wider range of tasks with similarly good results. This may be the first time that a biologically-based mechanism for controlling working memory has been demonstrated to compare favorably with the learning abilities of more abstract and biologically-implausible backpropagation-based temporal learning mechanisms. Other existing simulations of learning in the basal ganglia tend to focus on relatively simple sequencing tasks that do not require complex working memory maintenance and updating. Nevertheless, the central ideas behind the PBWM model are consistent with a number of these existing models (e.g., Schultz et al., 1995; Houk et al., 1995; Schultz et al., 1997; Suri et al., 2001; Contreras-Vidal & Schultz, 1999; Joel et al., 2002), thereby demonstrating that an emerging "consensus" view of basal ganglia learning mechanisms can be applied to more complex cognitive functions.



a) process event 1
(context gate closed)

b) update context
(context gate open)

c) process event 2
(context gate closed)

Figure 12: The simple recurrent network (SRN) as a gating network. When processing of each input event requires multiple cycles of settling, the context layer must be held constant over these cycles (i.e., its gate is closed, panel a). After processing an event, the gate is opened to allow updating of the context (copying of hidden activities to the context, panel b). This new context is then protected from updating during the processing of the next event, etc (panel c). In comparison, the PBWM model allows more flexible, dynamic control of the gating signal (instead of automatic gating each time step), with multiple context layers (stripes) that can each learn their own representations (instead of being a simple copy).

The central functional properties of the PBWM model can be summarized by comparison with the widely-used SRN backpropagation network, which is arguably the simplest form of a gated working memory model. The gating aspect of the SRN becomes more obvious when the network is updated iteratively for each input event (i.e., multiple cycles of updating are used per event, as in an interactive network, or to achieve reaction times from a feed forward network). In this case, it is clear that the context layer must be held constant and protected from updating during these cycles of updating (settling), and then it must be rapidly updated at the end of settling (Figure 12). Although the SRN achieves this alternating maintenance and updating by fiat, in a biological network it would almost certainly require some kind of gating mechanism. Once one recognizes the gating mechanism hidden in the SRN, it is natural to consider generalizing such a mechanism to achieve a more powerful, flexible type of gating. This is exactly what the PBWM model provides, by adding the following degrees of freedom to the gating signal: a) gating is dynamic, such that information can be maintained over a variable number of trials instead of automatically gating every trial; b) the context representations are learned, instead of simply being copies of the hidden layer, allowing them to develop in ways that reflect the unique demands of working memory representations (e.g., Rougier & O'Reilly, 2002); c) there can be multiple context layers (i.e., stripes), each with its own set of representations and gating signals. Although some researchers have used a spectrum of hysteresis variables to achieve some of this additional flexibility within the SRN, it should be clear that the PBWM model affords considerably more flexibility in the maintenance and updating of working memory information.

Although the PBWM model was designed to include many central aspects of the biology of the PFC/BG system, it also goes beyond what is currently known. For example, the specific role ascribed to the patch/striosome and SNc circuits provides testable hypotheses about the biology and function of these systems in the brain. We tested a large number of potential ideas about the function of this system in the context of an overall TD computation from the VTA system, and this was the only such idea that yielded computational improvements in performance. Therefore, it will be interesting to see if this idea stands up to further biological investigations.

Because the PBWM model represents a level of modeling intermediate between detailed biological models and powerful, abstract cognitive and computational models, it has the potential to build important bridges between these disparate levels of analysis. For example, the abstract ACT-R cognitive architecture has recently been mapped onto biological substrates including the BG and PFC (Anderson, Bothell, D., & Lebiere, submitted; Anderson & Lebiere, 1998), with the specific role ascribed to the BG sharing some central aspects of its role in the PBWM model. On the other end of the spectrum, biologically-based models have traditionally been incapable of simulating complex cognitive functions such as problem solving and abstract reasoning, which make extensive use of dynamic working memory updating and maintenance mechanisms to exhibit controlled processing over a time scale from seconds to minutes. The PBWM model should in principle allow models of these phenomena to be developed, and their behavior compared with more abstract models such as those developed in ACT-R. To meet this promise, more varied and rigorous tests of PBWM, combined with integration of relevant new biological data, will need to be undertaken.

Appendix: Implementational Details

The model was implemented using the Leabra framework, which is described in detail in O'Reilly and Munakata (2000) and O'Reilly (2001), and summarized here. See Table 3 for a listing of parameter values, nearly all of which are at their default settings. These same parameters and equations have been used to simulate over 40 different models in O'Reilly and Munakata (2000), and a number of other research models. Thus, the model can be viewed as an instantiation of a systematic modeling framework using standardized mechanisms, instead of constructing new mechanisms for each model. The model can be obtained by emailing the author at `oreilly@psych.colorado.edu`.

Pseudocode

The pseudocode for Leabra is given here, showing exactly how the pieces of the algorithm described in more detail in the subsequent sections fit together.

Outer loop: Iterate over events (trials) within an epoch. For each event:

1. Iterate over minus and plus phases of settling for each event.

```
   (a) At start of settling, for all units:
       i. Initialize all state variables (activation, v_m, etc).
       ii. Apply external patterns (clamp input in minus, input & output in plus).
   (b) During each cycle of settling, for all non-clamped units:
       i. Compute excitatory netinput ($g_e(t)$ or $\eta_j$, eq 9).
       ii. Compute kWTA inhibition for each layer, based on $g_i^\Theta$ (eq 13):
           A. Sort units into two groups based on $g_i^\Theta$: top $k$ and remaining $k+1$ to $n$.
           B. If basic, find $k$ and $k+1th$ highest; if avg-based, compute avg of $1 \rightarrow k$ & $k+1 \rightarrow n$.
           C. Set inhibitory conductance $g_i$ from $g_k^\Theta$ and $g_{k+1}^\Theta$ (eq 12).
       iii. Compute point-neuron activation combining excitatory input and inhibition (eq 7).
   (c) After settling, for all units:
       i. Record final settling activations as either minus or plus phase ($y_j^-$ or $y_j^+$).
2. After both phases update the weights (based on linear current weight values), for all connections:
   (a) Compute error-driven weight changes (eq 15) with soft weight bounding (eq 16).
   (b) Compute Hebbian weight changes from plus-phase activations (eq 14).
   (c) Compute net weight change as weighted sum of error-driven and Hebbian (eq 17).
   (d) Increment the weights according to net weight change.
```

Parameter	Value	Parameter	Value
E_l	0.15	$\overline{g_l}$	0.10
E_i	0.15	$\overline{g_i}$	1.0
E_e	1.00	$\overline{g_e}$	1.0
V_{rest}	0.15	Θ	0.25
τ	.02	γ	600
k In/Out	1	k Hidden	7
k PFC	4	k Matrix	1
k ImRew, NAc, Patch	3		
k_{hebb}	.01	ϵ	.01
to PFC k_{hebb}	.001*	to PFC ϵ	.001*

Table 3: Parameters for the simulation (see equations in text for explanations of parameters). All are standard default parameter values except for those with an *. The slower learning rate of PFC connections produced better results, and is consistent with a variety of converging evidence suggesting that the PFC learns more slowly than the rest of cortex (Morton & Munakata, 2002).

Point Neuron Activation Function

Leabra uses a *point neuron* activation function that models the electrophysiological properties of real neurons, while simplifying their geometry to a single point. This function is nearly as simple computationally as the standard sigmoidal activation function, but the more biologically-based implementation makes it considerably easier to model inhibitory competition, as described below. Further, using this function enables cognitive models to be more easily related to more physiologically detailed simulations, thereby facilitating bridge-building between biology and cognition.

The membrane potential V_m is updated as a function of ionic conductances g with reversal (driving) potentials E as follows:

$$\Delta V_m(t) = \tau \sum_c g_c(t)\overline{g_c}(E_c - V_m(t)) \quad (7)$$

with 3 channels (c) corresponding to: e excitatory input; l leak current; and i inhibitory input. Following electrophysiological convention, the overall conductance is decomposed into a time-varying component $g_c(t)$ computed as a function of the dynamic state of the network, and a constant $\overline{g_c}$ that controls the relative influence of the different conductances. The equilibrium potential can be written in a simplified form by setting the excitatory driving potential (E_e) to 1 and the leak and inhibitory driving potentials (E_l and E_i) of 0:

$$V_m^{\infty} = \frac{g_e \overline{g_e}}{g_e \overline{g_e} + g_l \overline{g_l} + g_i \overline{g_i}} \tag{8}$$

which shows that the neuron is computing a balance between excitation and the opposing forces of leak and inhibition. This equilibrium form of the equation can be understood in terms of a Bayesian decision making framework (O'Reilly & Munakata, 2000).

The excitatory net input/conductance $g_e(t)$ or η_j is computed as the proportion of open excitatory channels as a function of sending activations times the weight values:

$$\eta_j = g_e(t) = \langle x_i w_{ij} \rangle = \frac{1}{n} \sum_i x_i w_{ij} \tag{9}$$

The inhibitory conductance is computed via the kWTA function described in the next section, and leak is a constant.

Activation communicated to other cells (y_j) is a thresholded (Θ) sigmoidal function of the membrane potential with gain parameter γ:

$$y_j(t) = \frac{1}{\left(1 + \frac{1}{\gamma [V_m(t) - \Theta]_+}\right)} \tag{10}$$

where $[x]_+$ is a threshold function that returns 0 if $x < 0$ and x if $X > 0$. Note that if it returns 0, we assume $y_j(t) = 0$, to avoid dividing by 0. As it is, this function has a very sharp threshold, which interferes with graded learning learning mechanisms (e.g., gradient descent). To produce a less discontinuous deterministic function with a softer threshold, the function is convolved with a Gaussian noise kernel ($\mu = 0$, $\sigma = .005$), which reflects the intrinsic processing noise of biological neurons:

$$y_j^*(x) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-z^2/(2\sigma^2)} y_j(z - x) dz \tag{11}$$

where x represents the $[V_m(t) - \Theta]_+$ value, and $y_j^*(x)$ is the noise-convolved activation for that value. In the simulation, this function is implemented using a numerical lookup table.

k-Winners-Take-All Inhibition

Leabra uses a kWTA (k-Winners-Take-All) function to achieve inhibitory competition among units within a layer (area). The kWTA function computes a uniform level of inhibitory current for all units in the layer, such that the $k + 1$th most excited unit within a layer is generally below its firing threshold, while the kth is typically above threshold. Activation dynamics similar to those produced by the kWTA function have been shown to result from simulated inhibitory interneurons that project both feedforward and feedback inhibition (O'Reilly & Munakata, 2000). Thus, although the kWTA function is somewhat biologically implausible in its implementation (e.g., requiring global information about activation states and using sorting mechanisms), it provides a computationally effective approximation to biologically plausible inhibitory dynamics.

kWTA is computed via a uniform level of inhibitory current for all units in the layer as follows:

$$g_i = g_{k+1}^{\Theta} + q(g_k^{\Theta} - g_{k+1}^{\Theta}) \tag{12}$$

where $0 < q < 1$ (.25 default used here) is a parameter for setting the inhibition between the upper bound of g_k^{Θ} and the lower bound of g_{k+1}^{Θ}. These boundary inhibition values are computed as a function of the level of inhibition necessary to keep a unit right at threshold:

$$g_i^{\Theta} = \frac{g_e^* \bar{g}_e (E_e - \Theta) + g_l \bar{g}_l (E_l - \Theta)}{\Theta - E_i} \tag{13}$$

where g_e^* is the excitatory net input without the bias weight contribution — this allows the bias weights to override the kWTA constraint.

In the basic version of the kWTA function, which is relatively rigid about the kWTA constraint and is therefore used for output layers, g_k^{Θ} and g_{k+1}^{Θ} are set to the threshold inhibition value for the kth and $k + 1$th most excited units, respectively. Thus, the inhibition is placed exactly to allow k units to be above threshold, and the remainder below threshold. For this version, the q parameter is almost always .25, allowing the kth unit to be sufficiently above the inhibitory threshold.

In the *average-based* kWTA version, g_k^{Θ} is the average g_i^{Θ} value for the top k most excited units, and g_{k+1}^{Θ} is the average of g_i^{Θ} for the remaining $n - k$ units. This version allows for more flexibility in the actual number of units active depending on the nature of the activation distribution in the layer and the value of the q parameter (which is typically .6), and is therefore used for hidden layers.

Hebbian and Error-Driven Learning

For learning, Leabra uses a combination of error-driven and Hebbian learning. The error-driven component is the symmetric midpoint version of the GeneRec algorithm (O'Reilly, 1996), which is functionally equivalent to the deterministic Boltzmann machine and contrastive Hebbian learning (CHL). The network settles in two phases, an expectation (minus) phase where the network's actual output is produced, and an outcome (plus) phase where the target output is experienced, and then computes a simple difference of a pre and postsynaptic activation product across these two phases. For Hebbian learning, Leabra uses essentially the same learning rule used in competitive learning or mixtures-of-Gaussians which can be seen as a variant of the Oja normalization (Oja, 1982). The error-driven and Hebbian learning components are combined additively at each connection to produce a net weight change.

The equation for the Hebbian weight change is:

$$\Delta_{hebb} w_{ij} = x_i^+ y_j^+ - y_j^+ w_{ij} = y_j^+ (x_i^+ - w_{ij}) \quad (14)$$

and for error-driven learning using CHL:

$$\Delta_{err} w_{ij} = (x_i^+ y_j^+) - (x_i^- y_j^-) \quad (15)$$

which is subject to a soft-weight bounding to keep within the $0 - 1$ range:

$$\Delta_{sberr} w_{ij} = [\Delta_{err}]_+ (1 - w_{ij}) + [\Delta_{err}]_- w_{ij} \quad (16)$$

The two terms are then combined additively with a normalized mixing constant k_{hebb}:

$$\Delta w_{ij} = \epsilon [k_{hebb} (\Delta_{hebb}) + (1 - k_{hebb})(\Delta_{sberr})] \quad (17)$$

References

Alexander, G. E., DeLong, M. R., & Strick, P. L. (1986). Parallel organization of functionally segregated circuits linking basal ganglia and cortex. *Annual Review of Neuroscience*, *9*, 357–381.

Anderson, J. R., Bothell, D., D., B. M., & Lebiere, C. (submitted). An integrated theory of the mind. *Psychological Review*.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Baddeley, A. D. (1986). *Working memory.* New York: Oxford University Press.

Barch, D. M., Braver, T. S., Nystrom, L. E., Forman, S. D., Noll, D. C., & Cohen, J. D. (1997). Dissociating working memory from task difficulty in human prefrontal cortex. *Neuropsychologia*, *35*, 1373.

Barch, D. M., Carter, C. S., Braver, T. S., Sabb, F. W., MacDonald, A., Noll, D. C., & Cohen, J. D. (2001). Selective deficits in prefrontal cortex function in medication naive patients with schizophrenia. *Archives of General Psychiatry*, *58*, 280–8.

Braver, T. S., Barch, D. M., Keys, B. A., Carter, C. S., Cohen, J. D., Kaye, J. A., Janowsky, J. S., Taylor, S. F., Yesavage, J. A., & Mumenthaler, M. S. (2001). Context processing in older adults: Evidence for a theory relating cognitive control to neurobiology in healthy aging. *Journal of Experimental Psychology General*, *130*, 746–763.

Braver, T. S., & Bongiolatti, S. R. (2002). The role of frontopolar cortex in subgoal processing during working memory. *Neuroimage*, *15*, 523–536.

Braver, T. S., & Cohen, J. D. (2000). On the control of control: The role of dopamine in regulating prefrontal function and working memory. In S. Monsell, & J. Driver (Eds.), *Control of cognitive processes: Attention and performance XVIII* (pp. 713–737). Cambridge, MA: MIT Press.

Braver, T. S., & Cohen, J. D. (2001). Working memory, cognitive control, and the prefrontal cortex: Computational and empirical studies. *Cognitive Processing*, *2*, 25–55.

Charara, A., Heilman, C., Levey, A., & Smith, Y. (1999). Pre-and postsynaptic localization of GABA-B receptors in the basal ganglia in monkeys. *Neuroscience*, *95*, 127–140.

Cohen, J. D., Barch, D. M., Carter, C. S., & Servan-Schreiber, D. (1999). Schizophrenic deficits in the processing of context: Converging evidence from three theoretically motivated cognitive tasks. *Journal of Abnormal Psychology*, *108*, 120–133.

Cohen, J. D., Braver, T. S., & O'Reilly, R. C. (1996). A computational approach to prefrontal cortex, cognitive control, and schizophrenia: Recent developments and current challenges. *Philosophical Transactions of the Royal Society (London) B*, *351*, 1515–1527.

Contreras-Vidal, J. L., & Schultz, W. (1999). A predictive reinforcement model of dopamine neurons for learning approach behavior. *Journal of Comparative Neuroscience*, *6*, 191–214.

Cowan, R. L., & Wilson, C. J. (1994). Spontaneous firing patterns and axonal projections of single corticostriatal neurons in the rat medial agranular cortex. *Journal of Neurophysiology*, *71*, 17–32.

Dilmore, J. G., Gutkin, B. G., & Ermentrout, G. B. (1999). Effects of dopaminergic modulation of persistent sodium currents on the excitability of prefrontal cortical neurons: A computational study. *Neurocomputing*, *26*, 104–116.

Durstewitz, D., Kelc, M., & Gunturkun, O. (1999). A neurocomputational theory of the dopaminergic modulation of working memory functions. *Journal of Neuroscience*, *19*, 2807.

Durstewitz, D., Seamans, J. K., & Sejnowski, T. J. (2000a). Dopamine-mediated stabilization of delay-period activity in a network model of prefrontal cortex. *Journal of Neurophysiology*, *83*, 1733.

Durstewitz, D., Seamans, J. K., & Sejnowski, T. J. (2000b). Neurocomputational models of working memory. *Nature Neuroscience*, *3 supp*, 1184–1191.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Fellous, J. M., Wang, X. J., & Lisman, J. E. (1998). A role for NMDA-receptor channels in working memory. *Nature Neuroscience*, *1*, 273–275.

Frank, M. J., Loughry, B., & O'Reilly, R. C. (2001). Interactions between the frontal cortex and basal ganglia in working memory: A computational model. *Cognitive, Affective, and Behavioral Neuroscience*, *1*, 137–160.

Funahashi, S., Bruce, C. J., & Goldman-Rakic, P. S. (1989). Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *Journal of Neurophysiology*, *61*, 331–349.

Fuster, J. M. (1989). *The prefrontal cortex: Anatomy, physiology and neuropsychology of the frontal lobe, 3rd edition.* New York: Lippincott-Raven.

Fuster, J. M., & Alexander, G. E. (1971). Neuron activity related to short-term memory. *Science*, *173*, 652–654.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with lstm. *Neural Computation*, *12*, 2451–2471.

Goldman-Rakic, P. S. (1987). Circuitry of primate prefrontal cortex and regulation of behavior by representational memory. *Handbook of Physiology — The Nervous System*, *5*, 373–417.

Gorelova, N. A., & Yang, C. R. (2000). Dopamine d1/d5 receptor activation modulates a persistent sodium current in rats prefrontal cortical neurons in vitro. *Journal of Neurophysiology*, *84*, 75.

Graybiel, A. M., & Kimura, M. (1995). Adaptive neural networks in the basal ganglia. In J. C. Houk, J. L. Davis, & D. G. Beiser (Eds.), *Models of information processing in the basal ganglia* (pp. 103–116). Cambridge, MA: MIT Press.

Graybiel, A. M., & Ragsdale, C. W. J. (1978). Histochemically distinct compartments in the striatum of human, monkey, and cat demonstrated by acetylthiocholinesterase staining. *Proceedings of the National Academy of Sciences, USA*, *75*, 5723–5726.

Hernandez, P. J., Sadeghian, K., & Kelley, A. E. (2002). Early consolidation of instrumental learning requires protein synthesis in the nucleus accumbens. *Nature Neuroscience*, *5*, 1327–1331.

Hinton, G. E. (1989). Deterministic Boltzmann learning performs steepest descent in weight-space. *Neural Computation*, *1*, 143–150.

Hochreiter, S., & Schmidhuber, J. (1997). Long short term memory. *Neural Computation*, *9*, 1735–1780.

Houk, J. C., Adams, J. L., & Barto, A. G. (1995). A model of how the basal ganglia generate and use neural signals that predict reinforcement. In J. C. Houk, J. L. Davis, & D. G. Beiser (Eds.), *Models of information processing in the basal ganglia* (pp. 233–248). Cambridge, MA: MIT Press.

Joel, D., Niv, Y., & Ruppin, E. (2002). Actor-critic models of the basal ganglia: new anatomical and computational perspectives. *Neural Networks*, *15*, 535–547.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the 8th Confererence of the Cognitive Science Society* (pp. 531–546). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kelley, A. E., Smith-Roe, S. L., & Holahan, M. R. (1997). Response-reinforcement learning is dependent on N-methyl-D-aspartate receptor activation in the nucleus accumbens core. *Proceedings of the National Academy of Sciences, USA*, *94*, 12174–12179.

Koechlin, E., Corrado, G., & Grafman, J. (2000). Dissociating the role of the medial and lateral anterior prefrontal cortex in human planning. *Proceedings of the National Academy of Sciences*, *97*, 7651.

Kroger, J., Nystrom, L., O'Reilly, R. C., Noelle, D., Braver, T. S., & Cohen, J. D. (in preparation). Multiple levels of temporal abstraction in the prefronal cortex: converging results from a computational model and fmri.

Kubota, K., & Niki, H. (1971). Prefrontal cortical unit activity and delayed alternation performance in monkeys. *Journal of Neurophysiology*, *34*, 337–347.

Levitt, J. B., Lewis, D. A., Yoshioka, T., & Lund, J. S. (1993). Topography of pyramidal neuron intrinsic connections in macaque monkey prefrontal cortex (areas 9 & 46). *Journal of Comparative Neurology*, *338*, 360–376.

Lewis, B. L., & O'Donnell, P. (2000). Ventral tegmental area afferents to the prefrontal cortex maintain membrane potential 'up' states in pyramidal neurons via D1 dopamine receptors. *Cerebral Cortex*, *10*, 1168–1175.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*, 419–457.

Middleton, F. A., & Strick, P. L. (2000). Basal ganglia and cerebellar loops: Motor and cogntive circuits. *Brain Research Reviews*, *31*, 236–250.

Miller, E. K., Erickson, C. A., & Desimone, R. (1996). Neural mechanisms of visual working memory in prefontal cortex of the macaque. *Journal of Neuroscience*, *16*, 5154.

Mink, J. W. (1996). The basal ganglia: Focused selection and inhibition of competing motor programs. *Progress in Neurobiology*, *50*, 381–425.

Miyashita, Y., & Chang, H. S. (1988). Neuronal correlate of pictorial short-term memory in the primate temporal cortex. *Nature*, *331*, 68–70.

Morton, J. B., & Munakata, Y. (2002). Active versus latent representations: A neural network model of perseveration and dissociation in early childhood. *Developmental Psychobiology*, *40*, 255–265.

Movellan, J. R. (1990). Contrastive Hebbian learning in the continuous Hopfield model. In D. S. Touretzky, G. E. Hinton, & T. J. Sejnowski (Eds.), *Proceedings of the 1989 Connectionist Models Summer School* (pp. 10–17). San Mateo, CA: Morgan Kaufman.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*, 267–273.

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Computation*, *8*(5), 895–938.

O'Reilly, R. C. (1998). Six principles for biologically-based computational models of cortical cognition. *Trends in Cognitive Sciences*, *2*(11), 455–462.

O'Reilly, R. C. (2001). Generalization in interactive networks: The benefits of inhibitory competition and Hebbian learning. *Neural Computation*, *13*, 1199–1242.

O'Reilly, R. C., Braver, T. S., & Cohen, J. D. (1999). A biologically based computational model of working memory. In A. Miyake, & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control.* (pp. 375–411). New York: Cambridge University Press.

O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain.* Cambridge, MA: MIT Press.

Pennartz, C. M., Groenewegen, H. J., & Lopes da Silva, F. H. (1994). The nucleus accumbens as a complex of functionally distinct neuronal ensembles: an integration of behavioural, electrophysiological and anatomical data. *Progress in Neurobiology*, *42*, 719–761.

Plenz, D., & Kitai, S. T. (1998). Up and down states in striatal medium spiny neurons simultaneously recorded with spontaneous activity in fast-spiking interneurons studied in cortex-striatum-substantia nigra organotypic cultures. *Journal of Neuroscience*, *18*, 266–283.

Pucak, M. L., Levitt, J. B., Lund, J. S., & Lewis, D. A. (1996). Patterns of intrinsic and associational circuitry in monkey prefrontal cortex. *Journal of Comparative Neurology*, *376*, 614–630.

Robinson, A. J., & Fallside, F. (1987). *The utility driven dynamic error propagation network* (Technical Report CUED/F-INFENG/TR.1). Cambridge: Cambridge University Engineering Department.

Rougier, N. P., & O'Reilly, R. C. (2002). Learning representations in a gated prefrontal cortex model of dynamic task switching. *Cognitive Science*, *26*, 503–520.

Schmidhuber, J. (1992). Learning unambiguous reduced sequence descriptions. In J. E. Moody, S. J. Hanson, & R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems, 4* (pp. 291–298). San Mateo, CA: Morgan Kaufmann.

Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science, 275*, 1593.

Schultz, W., Romo, R., Ljungberg, T., Mirenowicz, J., Hollerman, J. R., & Dickinson, A. (1995). Reward-related signals carried by dopamine neurons. In J. C. Houk, J. L. Davis, & D. G. Beiser (Eds.), *Models of information processing in the basal ganglia* (pp. 233–248). Cambridge, MA: MIT Press.

Servan-Schreiber, D., Cohen, J. D., & Steingard, S. (1997). Schizophrenic deficits in the processing of context: A test of a theoretical model. *Archives of General Psychiatry, 53*, 1105–1113.

Stern, E. A., Kincaid, A. E., & Wilson, C. J. (1997). Spontaneous subthreshold membrane potential fluctuations and action potential variability of rat corticostriatal and striatal neurons in vivo. *Journal of Neurophysiology, 77*, 1697–1715.

Suri, R. E., Bargas, J., & Arbib, M. A. (2001). Modeling functions of striatal dopamine modulation in learning and planning. *Neuroscience, 103*, 65–85.

Surmeier, D. J., & Kitai, S. T. (1999). D1 and D2 modulation of sodium and potassium currents in rat neostriatal neurons. *Progress in Brain Research, 99*, 309–324.

Sutton, R. S. (1988). Learning to predict by the method of temporal diferences. *Machine Learning, 3*, 9–44.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction.* Cambridge, MA: MIT Press.

Wang, X.-J. (1999). Synaptic basis of cortical persistent activity: The importance of NMDA receptors to working memory. *Journal of Neuroscience, 19*, 9587.

Wickens, J. (1993). *A theory of the striatum.* Oxford: Pergamon Press.

Williams, R. J., & Zipser, D. (1992). Gradient-based learning algorithms for recurrent networks and their computational complexity. In Y. Chauvin, & D. E. Rumelhart (Eds.), *Backpropagation: Theory, architectures and applications.* Hillsdale, NJ: Erlbaum.