An Oscillatory Hebbian Network Model of Short-Term Memory

Ransom K. Winder*
rwinder@mitre.org
James A. Reggia
reggia@cs.umd.edu
Center for Advanced Study of Language and Department of Computer Science, University of Maryland, College Park, MD 20742, U.S.A.

Scott A. Weems
sweems@casl.umd.edu
Michael F. Bunting
mbunting@casl.umd.edu
Center for Advanced Study of Language and Department of Psychology, University of Maryland, College Park, MD 20742, U.S.A.

Recurrent neural architectures having oscillatory dynamics use rhythmic network activity to represent patterns stored in short-term memory. Multiple stored patterns can be retained in memory over the same neural substrate because the network's state persistently switches between them. Here we present a simple oscillatory memory that extends the dynamic threshold approach of Horn and Usher (1991) by including weight decay. The modified model is able to match behavioral data from human subjects performing a running memory span task simply by assuming appropriate weight decay rates. The results suggest that simple oscillatory memories incorporating weight decay capture at least some key properties of human short-term memory. We examine the implications of the results for theories about the relative role of interference and decay in forgetting, and hypothesize that adjustments of activity decay rate may be an important aspect of human attentional mechanisms.

1 Introduction

During the past several years, there has been growing interest in developing oscillatory neural network models for a variety of different tasks. These models consist of recurrent neural networks whose dynamics are characterized by persistent learned or designed rhythmic activity. For example,

*Ransom Winder is currently at MITRE Corporation.
oscillatory networks simulating biological central pattern generators have been studied both to investigate the mechanisms of locomotion in a number of animal species (Ijspeert, 2001) and as the basis for controlling movements of legged robots (Kimura, Fukuoka, Hada, & Takase, 2002). They have also been used to test hypotheses about the biophysical mechanisms underlying human electroencephalographic (EEG) activity (Nunez & Srinivasan, 2006) and for a variety of image processing tasks such as segmentation (Chen & Wang, 2002).

In this letter, we are concerned with the problem of developing oscillatory neural network models of short-term memory, the human memory system that retains information over brief time intervals (on the order of seconds) and has substantial capacity limitations, in contrast to the relatively limitless capacity of more permanent long-term memory (Baddeley, 2000; Cowan, 2001). For example, current evidence suggests that human short-term memory capacity is approximately four items (Cowan et al., 2005). In neurocomputational terms, long-term memories are typically viewed as being represented by the weights on connections in a neural network, while short-term memories are often represented by sustained activity patterns. Thus, in many neural models of associative memory, learned patterns are stored in a weight matrix. An initial network activity state, either assigned or determined by an input pattern and consisting of a partial or noisy activity pattern that is reasonably close to a previously stored memory, serves as a query as to the current contents of memory. This initial state evolves over time to a fixed-point attractor state (equilibrium state) that ideally corresponds to a recalled memory pattern that is active in short-term memory (Amit, 1989; Haykin, 1999).

While neural networks using fixed-point attractor states can be effective as memory models and have generated substantial theoretical and experimental analysis, they are typically limited to maintaining a single pattern at a time in short-term memory. Further, it is difficult to relate activity in these models to neurobiological systems where rhythmic activity, rather than fixed-point attractor states, is the rule (Buzsaki, 2006). In response to these and other concerns, a number of oscillatory memory models have been created and studied during the past several years. In these models, stored or recalled memory patterns are typically represented as rhythmic network activity in which multiple memory patterns are simultaneously active over the same neural substrate. This is possible because the network’s activity oscillates between activity states representing different stored patterns.

A remarkably diverse set of oscillatory memory models exists today. Some models are based on theories about the mechanisms underlying theta and gamma activity in specific brain regions such as the hippocampus or neocortex (Hasselmo, Bodelon, & Wyble, 2002; Ingber, 1995; Lisman & Idiart, 1995). For example, Koene and Hasselmo (2007, 2008) use a model of entorhinal cortex based on integrate-and-fire neurons that exhibit persistent spiking to show how theta rhythm may help explain order and capacity
effects in recall. Other models that also use individual spiking neurons are based on more abstract architectures (Raffone & Wolters, 2001), while still others have adopted a higher-level approach such as Wilson-Cowan oscillators (Chakravarthy & Ghosh, 1996; Hayashi, 1994; Wang, 1995). Many of these past oscillatory models are quite complex (see Howard, in press, for a review).

A particularly simple and elegant approach to creating oscillatory short-term memory models is based on minimally modifying Hebbian associative memories having fixed-point attractor states so that they become oscillatory. In the following, we refer to all such models as simple oscillatory memories. For example, Horn and Usher (1991) produced a simple oscillatory memory by introducing dynamic thresholds into Hopfield networks (Hopfield, 1982; Amit, 1989). With this approach, whenever a node has a particular activity level $\pm 1$, the threshold of that node gradually changes so that eventually the node switches its activity level to the complementary value. When such a network is presented with an input that is a superposition of multiple stored memories, it is found to oscillate between activity states that represent these individual memories, thereby indicating its recognition and recall of the memories in parallel. Similar behaviors have been produced based on Hopfield networks modified to use dynamic synapses (Pantic, Torres, Kappen, & Gielen, 2002) or negative feedback with asymmetric connection weights (Brown & Collins, 2000).

Simple oscillatory memories derived from Hopfield networks are intriguing in their simplicity as models of short-term memory. In this letter, we present a simple oscillatory memory based on dynamic thresholds as used by Horn and Usher (1991), except that with our approach, the model is extended to include rapid decay of connection weights. This weight decay allows the network to have a dependency on the order in which it sees input stimuli, something that is not the case with classical Hopfield networks. Further, it lets us examine the relative roles of interference and decay as mechanisms underlying forgetting. To evaluate the model, we use data that we collect from human subjects performing a running memory span task. This task involves rapidly presenting a sequence of stimuli that are to be recalled subsequently by the subject. We show that our model can demonstrate recall performance similar to the behavioral data that we obtain: a capacity limit of approximately three items and a prominent recency effect. To our knowledge, no previous work has examined how the performance of simple oscillatory memories compares to real-world behavioral data collected from human subjects on a short-term memory task as we do here.

2 Methods

2.1 Model Description. Short-term memory is modeled using a fully connected network of 35 linear threshold nodes similar to many past neural network models. There are two possible values for the activity state $a_i$ of
Figure 1: Three examples of the encodings used for network memory states, where the network’s 35 nodes are pictured as a $7 \times 5$ array of elements for illustrative purposes. Nodes with an activation level of $+1$ are indicated in black, and those with an activation level of $-1$ are in white. From left to right, these examples represent P, W, and H.

Each node $i$, $-1$ and $1$, and these values change over time, governed by the activity state of the network and connection strengths, as described below. Memory patterns to be stored are essentially arbitrary. However, to facilitate visual interpretation of the network’s state, the nodes in the network are displayed as a $7 \times 5$ array, and the memory state $\vec{a}$ is taken to correspond to a specific letter (A–Z), each represented so as to pictorially resemble the letter visually, as illustrated in Figure 1.

Connection strengths for this 35-node network are kept in a $35 \times 35$ weight matrix $W$, where each weight $w_{ij}$ is a real-valued number. Connection strengths between two nodes are the same in both directions ($w_{ij} = w_{ji}$ everywhere), so $W$ is symmetric. Node activation levels $a_i$ are updated according to the activation rule

$$a_i = \begin{cases} 
+1 & \text{with probability } (1 + e^{-h_i/T})^{-1} \\
-1 & \text{with probability } (1 + e^{+h_i/T})^{-1},
\end{cases}$$

with the input $h_i$ to node $i$ being given by

$$h_i = \sum_j w_{ij}a_j - \theta_i - K_i,$$

where $\theta_i$ is the threshold associated with node $i$, $K_i$ is a biasing factor that compensates for the unequal numbers of $+1$ and $-1$ values among the memory patterns,$^1$ and $T$ is a temperature parameter (0.1 in our simulations). Note that $h_i$, $a_i$, $w_{ij}$, $\theta_i$, and $K_i$ are all functions of time.

---

$^1$Based on equation 2.10 in Horn and Usher (1991), factor $K_i$ was computed as

$$K_i = \sum_p A \cdot M \cdot a_p^\lambda + 2(M - A),$$

where $A$ is the average activity level of all nodes over all 26 letter patterns $\lambda$ that were used, and $M$ is the average activity of the network at a given time step. $A = -0.147$ for our letter stimuli.
Following Horn and Usher (1991), and unlike in fixed-point attractor networks where after learning a node’s threshold is fixed, the threshold values here are dynamic, changing with each time step $t$ according to

$$
\theta_i(t) = k_r R_i(t)
$$

$$
R_i(t + 1) = \frac{R_i(t)}{c} + a_i(t + 1),
$$

where $R_i(0) = 0$ and $c > 1$. Thus, for example, when the activation level $a_i = +1$, this causes the threshold for node $i$ to rise slowly, making it more likely that the node will become negative during the next time step. Similarly, when the value of the node is $-1$, the threshold drops, making it more likely that the node will become positive. This way, when the network is run for any length of time, the network state can oscillate and explore different patterns stored in its weight matrix. We used the values $k_r = 0.15$ and $c = 1.2$ in the simulations described below.

To simulate the presentation of a sequence of letters that are being consecutively stored in a subject’s short-term memory during a running memory span task, $W$ is initialized with $w_{ij} = 0$ for all $i$ and $j$, and then a sequence of memory states corresponding to that sequence of letters is imposed on the network. As each presented memory state is transiently present, the connection strengths $w_{ij}$ are all concurrently updated according to the weight change rule,

$$
w_{ij} = (1 - k_d)w_{ij}^{old} + \frac{1}{N}a_i a_j (1 - \delta_{ij})
$$

where $N$ is the number of nodes in the network, $k_d$ is a decay rate capturing how weights diminish over time ($0 \leq k_d < 1$), and $\delta_{ij}$ is Kronecker’s delta (the latter ensures $w_{ii} = 0$ for all $i$, so weights on self-connections are fixed at zero). For the computational experiments that follow, we used $N = 35$, while $k_d$ values vary between different simulations. The second term on the right side of this weight change rule implements Hebbian and correlational weight changes as in many past neural network models of memory, including that of Horn and Usher (1991). However, our weight change rule as a whole differs in explicitly incorporating a weight decay factor $-k_d w_{ij}$ that gradually reduces the influences of old memory patterns. It still produces a symmetric weight matrix $W$ with zero entries on the main diagonal. If relatively few memory patterns are in memory at any point in time and the decay rate is small, with more traditional Hopfield networks having constant thresholds, the stored memories would typically tend to be fixed-point attractor states (energy minima). Thus, when our model starts in an arbitrary initial state $\vec{a}$, its activity state would be expected to change until it reaches one of these stored memory states, but would then not remain
fixed in that state due to the dynamic thresholds as explained above. This leads to oscillatory behavior.

### 2.2 Measuring Stimulus Patterns Actively Retained in Short-Term Memory

To assess how well the network’s current activity state $\vec{a}$ matches the activity state $\vec{a}_\lambda$ corresponding to one of the 26 specific letter stimuli $\lambda$, we first compute a measure of the distance $d_\lambda$ between $\vec{a}$ and $\vec{a}_\lambda$ for that letter as

$$d_\lambda = \frac{1}{2} \sum_{i=1}^{N} |a^\lambda_i - a_i|.$$  

This measure is essentially the same as the Hamming distance between two binary numbers. The similarity $s_\lambda$ of current state $\vec{a}$ to stimulus pattern $\vec{a}_\lambda$ is then computed based on this distance measure as

$$s_\lambda = 0.85^d_\lambda,$$

which lies between 0.0 and 1.0. A value $s_\lambda = 1.0$ at any time step indicates a perfect match between the current state $\vec{a}$ and the stimulus pattern $\vec{a}_\lambda$, while progressively lower values of $s_\lambda$ indicate progressively worse matches. We use this nonlinear measure of similarity rather than a linear measure such as $N - d_\lambda$ to accentuate differences between activity states representing similar letters such as I and T that have substantial overlap.²

The process of testing for the retention of specific letters that have just been presented sequentially to the model is done as follows. The network is started in a random initial activity state $\vec{a}$, and then the network’s state is allowed to evolve according to the dynamics described above for a 200 time step test period. Because of the changing nature of the thresholds, the network does not reach a fixed state during the test period, but instead typically oscillates between states that are at or close to some of the activity patterns $\vec{a}_\lambda$ that were shown to it during the simulated running memory span task. During this testing period, the similarity measure $s_\lambda$ for each of the stimuli $\lambda$ is recorded at each time step.

In the simulations that follow, we label a specific stimulus or letter $\lambda$ that was presented to the network as being actively present in memory, and thus recalled by the model, only if $s_\lambda$ reaches a value of 1.0 during the 200 time step testing interval. This means the stimulus must be perfectly

---

²The value 0.85 is somewhat arbitrary (0.75 or 0.90 or other similar values would serve as well). We selected 0.85 prior to systematic data collection by inspecting pilot simulation results because this value gave visually evident separation of retained versus “forgotten” memory stimuli when plotted (like those plotted in Figure 2). Note that any of these values give $s_\lambda = 1$ when $d_\lambda = 0$, so the specific value used for this parameter does not alter in any way whether a memory pattern is counted as recalled or not in our results.
recalled by the network during this testing period at least once. We use this strict criterion because the similarity measure of some letters that are similar to each other can tend to rise and fall synchronously. For example, when the current state \( \vec{a} \) is an exact match for the letter T (i.e., state \( \vec{a}^T \) representing letter T is present with \( s_T = 1.0 \)), then letter I would have a small but significant \( s_I \) value simultaneously due to its overlap with T.

2.3 Behavioral Data. We collected behavioral data on a running memory span task for comparison with the model’s performance, roughly following the designs of Pollack, Johnson, and Knaff (1959) and Bunting, Cowan, and Saults (2006). Our human experimental data were obtained from 38 adult subjects (13 females, 25 males, mean age 25) who were shown a rapidly presented, two per second sequence of 12 to 20 randomly ordered stimuli under computer control and were asked to remember the most recent six items in the order of their presentation.\(^3\) Subjects indicated the stimuli that they recalled by clicking on a subsequent graphical display of all possible stimuli. Recall was measured by assessing accuracy of recall as a function of stimulus position. A stimulus was counted as accurately recalled only if (1) it was presented in the retention window (i.e., the last six items, depending on instructions), (2) it was correctly recalled by the participant, and (3) it was recalled in the same position as it was presented (counting backward from the final, most recent stimulus). Any item presented prior to the retention window that was recalled was considered a false positive, as was any item that was not presented at all but was recalled. Any item from the retention window that was not recalled was considered a miss. Any item that was presented in the retention window but recalled in the incorrect position was also counted as wrong (e.g., if the last six items presented were “1 2 3 4 5 6” and the subject recalled “4 3 2 6 5 1,” then only “5” was counted as correct). Twelve trials were conducted for the task, with each subject requiring roughly 20 minutes per subject. No time restrictions were placed on subject responses. All 38 subjects completed the task.

3 Results

Figure 2 shows a representative example of how, following presentation of a sequence of letters, network activity measured using \( s_\lambda \) oscillates between

\(^3\)As the stimuli are meaningless patterns to the computational model, numerical digits were used as stimuli with human subject testing rather than letters to allow simple and efficient behavioral response entry using a mouse. Subjects clicked on a graphical display of the digits 1 through 9 organized in the style of a telephone keypad. In testing human subjects, a digit was never repeated within a moving window of seven stimuli, and never more than twice. We do not expect that digit repetition significantly affected responses because careful attention was paid to avoiding any contamination from strategies that could improve memory capacity, minimizing interference from repetition within any given trial relative to that throughout the experiment.
Figure 2: Values of $s_\lambda$ during a sample run of the network over 200 time steps starting from a random initial state after being presented with the temporal sequence of eight stimuli M, L, X, N, E, F, H, and B in that order. Stimuli E, H, and B are labeled as being recalled in this case, as their oscillations peak during this period at activity states that perfectly match the stimuli.
Figure 3: Plot over time of when the values of $s_\lambda$ reached their peaks for the eight stimuli during the same example run as in Figure 2. Solid black marks indicate when $s_\lambda$ reached the maximum possible value of 1.0, while gray marks indicate when $s_\lambda$ exceeded 0.8 but did not reach 1.0. By viewing the oscillation peaks together like this, one can see that the oscillatory states rotate between the three recalled memory patterns for the fifth, seventh, and eighth stimuli (E, H, and B). Only the middle 100 time steps of Figure 2 are shown here, expanded horizontally.

memory states of some of the presented letters, indicating the retention of those letters in short-term memory. In this case, where a decay rate of $k_d = 0.2$ is used, the sequence of stimuli from first to last was M, L, X, N, E, F, H, and B. Subjectively, one can see in Figure 2 that the oscillations associated with the earlier letters have relatively small amplitudes (only partial matches), while those of the more recently presented letters are more prominent. Using our criterion that a letter $\lambda$ is retained in short-term memory (recalled) if and only if its activity pattern $\vec{a}_\lambda$ occurs exactly ($s_\lambda = 1.0$) during the test period, the letters E, H, and B would be labeled as recalled in this example.

Figure 3 shows when the oscillations in $s_\lambda$ values peaked at 0.8 or above for the eight stimuli of this same example during just the middle of the testing period. Peaks in the oscillations associated with the different recalled letters (E, H, and B) alternate with each other, allowing the three remembered letters to be retained in short-term memory simultaneously, unlike with fixed-point neural associative memories. Note that the oscillations are irregularly spaced and not periodic.

Figure 4 displays the fraction of letters recalled (maintained as oscillatory states in short-term memory) for stimuli in each position, averaged over 1000 random input sequences of eight different stimuli. When there is no decay ($k_d = 0$), the fraction of letters recalled is largely independent of a letter’s position in the input sequence, resulting in a flat curve. This result can be related to fixed-point attractor networks where the final weights,
and hence network performance as an associative memory, are independent of the order in which the input patterns are stored. Further, recall of any observed letter in this case is quite poor, as would be expected; for a network of the size used here, the number of stimuli used far exceeds the expected memory capacity of an equal size fixed-point attractor associative memory (Hopfield, 1982; Amit, 1989). With no decay, the interference between the stored memory patterns is excessive, preventing almost all letters from being retained effectively in short-term memory. In contrast, when the decay rate is very large ($k_d = 0.5$), a very steep curve is seen (see Figure 4), with the single most recently presented (final) letter always being retained. This occurs because the weight changes from previously stored stimulus patterns quickly dissipate, and even recently presented earlier stored patterns no longer interfere significantly with the final letter’s retention. In this case, the fraction of the first six presented letters that are recalled is almost zero, reflecting that they have been erased from memory. This can be contrasted with the roughly 20% recall rate of presented letters when there is no decay at all. Intermediate behaviors are seen for intermediate values of the decay rate, as is shown for $k = 0.1$ in Figure 4.

Figure 5 shows, for each decay rate value used in the results of Figure 4, the mean total number of stimuli recalled (memory capacity), averaged over the 1000 stimuli sequences tested. These results show that memory capacity is highest with moderate, intermediate decay rates. If decay is very low, older input stimuli are substantially retained, and the attempt to store an excessive number of memory patterns results in too much interference. If decay is too high, even recently observed stimuli are lost. Maximum recall

---

Figure 4: Fraction of letters recalled versus stimulus position when using different decay rates $k_d$. Each curve plots the fraction of presented letters in each position of eight-letter sequences that were recalled correctly during the testing period following presentation of the stimuli, averaged over 1000 trials using the same decay rate. Stimulus 1 was the first stimulus presented and stimulus 8 the last.
Figure 5: Mean number of letters recalled for different decay rates. This peaks at a memory capacity of roughly 2.5 for intermediate decay rate values (around 0.15).

of observed stimuli occurs when there is a balance between information lost due to decay and interference. The appendix discusses the issue of why an intermediate value of the decay rate tends to maximize memory capacity from a more analytic point of view.

How does position-specific recall by the model compare with that exhibited by human subjects? The curves in Figure 6 display experimental results when the human subjects are instructed to remember the six most recent stimuli. As also plotted in Figure 6, the model is able to approximate closely the results found with the human subjects when it is run with a 0.1 decay rate with input sequences of length six. The mean total memory capacity for recall of six stimuli was 2.73 items for human subjects versus 2.69 items for the model. Thus, both the model’s total memory capacity and its position-specific stimulus retention patterns were in close agreement with those seen with the human subjects.

As described in section 2, human subject data were collected under conditions where the subjects were presented with stimulus sequences of 12 to 20 items and were told to recall only the last 6, regardless of the actual sequence length. To explore how the model was affected by seeing more stimuli than were to actually be recalled, we repeated the simulations as above but now using 20 stimuli in each sequence, even though recall of only the last 6 stimuli was of interest. Figure 7 shows how the model’s performance in position-specific recall can, with an adjusted decay rate, still reasonably match the human subject results. The model results here are again averaged over 1000 different random sequences of letter stimuli. We found that a modestly higher decay rate (0.185) provides an approximate match to the six-back human data. The model’s mean total memory capacity for recall of six stimuli was now 2.28 items (versus 2.73 items for human subjects).
4 Discussion

In this work we introduced a simple oscillatory memory model of short-term memory, examined some of its properties, and compared its behavior to that of human subjects on a running memory span task. Our model’s dynamics are intrinsically oscillatory due to the use of rapidly varying threshold values, and recall of an item is dependent on the time elapsed since it was observed due to the use of rapidly decaying weights. Unlike with many past neurocomputational models of memory, we assessed recall by initializing the model’s activity to a random state rather than by initializing it to a noisy or partial stored memory pattern, or by biasing the network’s dynamics by applying an external input pattern that represents a noisy or partial stored pattern. We found that when moderate decay rates were used, this approach resulted in a short-term memory capacity of two to three items, a value that is comparable to what has been observed in experimental studies by others (Baddeley, 2000; Cowan, 2001; Cowan et al., 2005) and that matches the memory capacity that we observed in a group of human subjects performing a similar running memory task. The model
Figure 7: Comparison of the position-specific fraction of letters recalled by the model to the same measures for human subjects. Unlike with the data shown in Figure 6, the model is now presented with 20 stimuli, but (for comparison to the human behavioral data) only the accuracy of the most recent six stimuli is plotted. A somewhat higher decay value of $k_d = 0.185$ is used that produces a similar curve to what is observed with the human subjects, and the model again exhibits greater recall of more recent stimuli.

It should also be noted here that this model is intended to simulate short-term memory processing only, and so it is not intended to address any processes by which semantic or other long-term memory information is accessed to aid storage or recall. Indeed, it has been well established that short-term memory capacity is higher for familiar items for which long-term representations exist, compared to novel stimuli (Hulme, Maughan, & Brown, 1991; see Cowan, 2001, for review). This benefit is likely due to the fact that for novel stimuli, representations must be created before retention can successfully occur. As the model made no assumptions regarding the relationship between short-term and long-term memory, the choice of letters as the retained stimuli seemed a reasonable assumption. Still, we recognize that more complex models of short-term memory could benefit from expansion by making predictions regarding the important role of long-term memory for temporarily retained information.
Also, although the computational model is oscillatory in nature (though not periodic), it is not intended to make any predictions regarding frequency-specific responses actually produced in the brain during short-term memory retention. There is a rich and expanding research literature showing that EEG activity in the theta band plays a key role in the maintenance of temporarily retained information (Jensen & Tesche, 2002) and that hippocampal structures play a significant role in this frequency component (Kahana, Deelig, & Madsen, 2001; Rizzuto et al., 2003). There has also been recent research to show that higher-frequency oscillatory activity (i.e., gamma response) increases approximately linearly with increased memory load (Howard et al., 2003), and that greater gamma activity during encoding predicts greater likelihood of later recall (Sederberg, Kahana, Howard, Donner, & Madsen, 2003). The results of our model are promising in suggesting that oscillatory neural models can show similar capacity limitations as with humans, but they do not allow us to make predictions regarding frequency-specific contributions to EEG, especially as the model oscillations recorded are in terms of the extent to which a specific distributed memory pattern is present (quantity $\lambda$) and not in terms of amount of network activity. While the model does retain information about stimulus order—indeed, stimulus order effects emerge in the model as a result of gradual decay—it does not address issues of temporal sequencing. As recent single-unit recording evidence suggests that ordered sequences of activation are observed in rats (Foster & Wilson, 2006), this may be an interesting area of future expansion of the model.

Our study adds to a rapidly growing literature on computational models of short-term memory by examining the role of weight decay on simple oscillatory memories. Many past models of short-term memory have employed lateral inhibition between representational units to establish competition between activated entities, and thus capacity limitations. For example, Haarmann and Usher (2001) present a model of semantic short-term memory that functions in this fashion. Our approach differs in not explicitly building in such lateral inhibition (although inhibitory weights do occur during pattern storage), with competition between memory patterns arising in the dynamics due to the interference occurring between the nonorthogonal memory patterns. Other recent models of short-term memory, inspired by specific neuroanatomical structures, have used separate modules for memory representation, maintenance, and selective gating. For example, Frank, Loughry, and O’Reilly (2001) and O’Reilly and Frank (2006) incorporate modules representing prefrontal cortex and basal ganglia, with the latter modulating which sensory stimuli are kept active. Our approach does not use a complex architecture or gating mechanisms, and thus shows that some basic behavioral properties of human short-term memory (limited memory capacity, recency effect, and shifts in position-specific stimulus recall) can be captured by a surprisingly simple neurocomputational mechanism. Still other recent short-term memory models have been based on
modulation of persistent neuronal firing by rhythmic changes to membrane potential at theta frequencies (Koene & Hasselmo, 2007, 2008). Our approach is quite different in that storage is based primarily on synaptic connectivity, and memory capacity limitations arise mainly due to synaptic decay and pattern interference. However, it is an interesting question as to whether the dynamic thresholds in our model might correspond to the changes in effective thresholds brought about by the modulating theta activity in these more physiologically realistic implementations.

Perhaps the most interesting finding with the model is that by adjusting just the weight decay rate, one can produce shifts in the model’s memory capacity and position-specific recall rates, as is demonstrated in Figure 4. This represents a prediction of the model that by adjusting the decay rate, one could reasonably match the shifts exhibited by human subjects who were instructed to recall different-length stimuli sequences. This would be especially remarkable given the simplicity of our model and that it requires adjustment of only a single parameter. This prediction relates to long-standing issues in the cognitive science literature concerning the nature of forgetting. For example, one view of forgetting is that short-term memory is subject to decay (Brown, 1958), while an alternative view is that forgetting is due to interference between competing elements that are simultaneously vying for attention (Waugh & Norman, 1965). Our model incorporates both interference and decay as mechanisms for forgetting and shows that the latter can partially mitigate effects from interference, consistent with evidence in past behavioral studies (Altmann & Gray, 2002).

In general, it is difficult to map processes in neurocomputational models to cognitive processes, but sometimes there are analogs that are worth considering. The observation that adjustments to decay rate control not only the total short-term memory capacity (see Figure 5) but also position-specific stimulus recall rates (see Figure 4) raises the issue of whether altering decay rate might be a useful mechanism permitting a cognitive system to control short-term memory characteristics. Specifically, our model is consistent with the hypothesis that dynamic adjustments to activity decay rate may be an important aspect of the human attention mechanisms that control forgetting (Altmann & Gray, 2002).

It is already well established that attention is a cognitive property that can be manipulated based on the needs of the task at hand (Broadbent, 1982; Downing & Pinker, 1985; Eriksen & St. James, 1986) and that attentional scope can be adjusted during visual search and memory recall between being more focused or more diffuse (Engle, 2002; Kane & Engle, 2002). Based on our modeling results, we hypothesize that altering the decay rate could serve as a means by which attentional mechanisms could act to manipulate attentional scope. More focused attention is simulated in the model by a higher decay rate, so that attention is directed more intently on a smaller number of items. In this way, decay is used as a means for combating proactive interference, with higher decay rates leading to more
effective retention of recent information, but also at the expense of that which was presented before it.

For the running memory span task used here involving rapid presentation of stimuli, human subjects attempt to hold presented stimuli in a limited-capacity memory without the use of rehearsal (Bunting et al., 2006). Assuming that maintaining such stimuli depends on attentional resources, then changing instructions requiring subjects to retain varying numbers of stimuli (i.e., not just six as we did in our behavioral experiments) would be expected to have a great effect. Specifically, if attention is drawn sufficiently thin so that activation maintenance is small across all retained stimuli (a low decay rate in our model), then with longer stimulus sequences (e.g., a task requiring human subjects to recall 12 stimuli) few or none of the stimuli would be expected to retain activation levels above some cognitive threshold required for successful recall due to interference, although no doubt some attenuated recency effect will still be present. This is both a surprising and informative prediction from the model, and it suggests that overloading subjects’ attentional resources (i.e., drawing attention sufficiently thin) has a detrimental effect on retention. Future behavioral testing with a varying-length recall task could therefore either refute or strongly support the model we have presented here.

To our knowledge, the work reported here is the first comparison of simple oscillatory model properties to human behavioral data. While the results are encouraging, they leave open a number of issues that should be examined in future work. Perhaps the most pressing issue concerns the generality of these results. It will be important to determine whether similar correspondences between model properties and human behavior can be produced as readily with other stimulus sets, while varying the number of stimuli that subjects are instructed to retain and recall, and across other short-term memory tasks. For example, it would be useful to compare the model’s results against human data using unfamiliar or abstract symbols or nonwords where subject performance is based on free recall. Similarly, with the model, it would be useful to examine how long-term memory influences the results of recall. Another future issue, not examined in our work, would be to characterize model behavior in the absence of interference using orthogonal stimulus patterns, allowing the effects of weight decay to be studied in isolation.

Appendix: Analysis of the Effects of Varying Rate of Decay

Given the nonlinear, stochastic activation dynamics of our model and the use of dynamic thresholds, it is difficult to analytically predict an exact value of the decay rate $k$ that will maximize memory capacity. However, following Horn and Usher (1991), we can view the network’s activation pattern as flowing toward what would be fixed points (ideally the stored
memories at energy minima) in a traditional nonoscillatory Hopfield model, from which it is repeatedly destabilized by the dynamic threshold. From this perspective, we can ignore the dynamic thresholds that produce the oscillatory dynamics (i.e., we assume fixed thresholds of zero) and focus on determining which value of $k$ will maximize how many of $M$ sequentially learned patterns $S = \{\tilde{a}^1, \tilde{a}^2, \ldots, \tilde{a}^M\}$ are stored in a network having $N$ nodes when weight decay is being used. Representing each memory pattern $\tilde{a}^p$ as a column vector, our weight change procedure for acquiring $w_{ij}$ (see section 2) can be written in matrix form as

\begin{align*}
\text{initialize } W &= 0 \\
\text{for } p = 1, 2, \ldots, M \text{ do } W &= (1 - k)W + \frac{1}{N}(\tilde{a}^p(\tilde{a}^p)^T - I). \\
\end{align*}

This learning procedure produces a resultant weight matrix,

$$W = \frac{1}{N} \sum_{p=1}^{M} (1 - k)^{M-p}(\tilde{a}^p(\tilde{a}^p)^T - I),$$

that is symmetric with all zero values on the main diagonal.

Once $W$ is learned in this fashion, if the network starts in the $q$th stored memory state $\tilde{a}^q$, the inputs $\tilde{h}^q_i$ to individual nodes $i$ are given by elements of

$$\tilde{h}^q = W\tilde{a}^q = \left[(1 - k)^{M-q} - \frac{1}{N} \sum_{p=1}^{M} (1 - k)^{M-p}\right] \tilde{a}^q + \frac{1}{N} \sum_{\substack{p = 1 \\ p \neq q}}^{M} (1 - k)^{M-p}\tilde{a}^p(\tilde{a}^p)^T \tilde{a}^q.$$

Here we have moved the term $\frac{1}{N}(1 - k)^{M-q}\tilde{a}^q(\tilde{a}^q)^T \tilde{a}^q = (1 - k)^{M-q}\tilde{a}^q$ in the summation on the right to be the left-most term in $\tilde{h}^q$. In general, computing the right-most summation in $\tilde{h}^q$ is complicated by the fact that it depends on the similarity of all of the patterns $\tilde{a}^p$ in $S$ to $\tilde{a}^q$ as measured by their dot products $A_{pq} = \tilde{a}^p : \tilde{a}^q = (\tilde{a}^p)^T \tilde{a}^q$. However, we can give a rough ballpark estimate for this summation if we replace the individual $\tilde{a}^p$ and $A_{pq}$ values with their means $\tilde{a}_\mu = \frac{1}{n} \sum_p \tilde{a}^p$ and $A_\mu = \frac{1}{n(n-1)} \sum_p \sum_{q \neq p} A_{pq}$, respectively, where the average is taken over the set of $n$ possible memory patterns from which those in a specific set $S$ are drawn. These means are easily computed with a one-time calculation from finite data sets like the
one that we used in this study. This gives the approximation

$$\vec{h}^q \approx \left[ (1 - k)^{M-q} - \frac{1}{N} \sum_{p=1}^{M} (1-k)^{M-p} \right] \vec{a}^q + \left[ \frac{A_\mu}{N} \sum_{p=1 \atop p \neq q}^{M} (1-k)^{M-p} \right] \vec{a}_\mu.$$

We can use this $\vec{h}^q$ estimate to see why an intermediate value of $k$ optimizes memory capacity. First note that $\vec{h}^q$ has the form $\vec{h}^q = F_q \vec{a}^q + F_\mu \vec{a}_\mu$, where $F_q$ and $F_\mu$ are scalar functions of $k$. The first self-support term $F_q \vec{a}^q$, considered in isolation, tends to make $\vec{a}^q$ a fixed point (in the low-temperature limit) and thus an energy minimum, if $F_q$ is large. Assuming that $N$ is relatively large compared to $M$, as is typically the case, then the first term $(1 - k)^{M-q}$ will generally be the most important factor; this term represents the effects of decay on recall of stored pattern $\vec{a}^q$. The second interference term $F_\mu \vec{a}_\mu$, which represents cross-talk due to other stored patterns, tends to prevent $\vec{a}^q$ from being a fixed point if $F_\mu$ is large. The magnitude of $F_\mu$, and thus the amount of interference, depends on how similar the memory patterns are to one another as measured by the $A_{pq}$ values. Thus, for a specific $\vec{a}^q$, we have a multiobjective optimization problem: find a value of $k$ that maximizes $F_q$ while simultaneously minimizing $F_\mu$, subject to the constraint $0 \leq k < 1$. Such a $k$ value will maximize the probability that $\vec{a}^q$ is stored in memory (i.e., is a fixed point). However, as in many multiobjective optimization problems, there is a trade-off: a small value of $k$ will have the desirable effect of maximizing $F_q$ and hence self-support (which, assuming $N$ is large, is primarily determined by $(1 - k)^{M-q}$), but will also have the deleterious effect of simultaneously maximizing $F_\mu$, and hence of maximizing interference. The opposite holds for a large value of $k$. It is this trade-off between self-support and interference that ultimately makes an intermediate value of $k$ optimal.

Solving multiobjective optimization problems like ours that are subject to constraints is generally recognized as a very difficult problem (Deb, 2001). However, we can estimate a reasonably good $k$ value that will maximize memory capacity for input sequences $S$ of length $M$ by simply estimating the mean of the values $k$ that maximize the ratio $F_q/F_\mu$, subject to the constraint $0 \leq k < 1$, for each of the memory patterns $\vec{a}^q$ in $S$, averaged over the patterns in $S$. For each $\vec{a}^q$, a preferred $k$ value is easily estimated by identifying the maximum $F_q/F_\mu$ value as $k$ is sampled between 0 and 1. For example, for the conditions under which memory capacity was measured in Figure 5 ($N = 35$ nodes, $M = 8$ patterns, and using the $n = 26$ letter patterns for which one can compute $A_\mu = 7.04$), this approach predicts an optimal value of $k = 0.22$, which is reasonably close to the measured value of approximately 0.15 (see Figure 5) given the approximations made above in deriving $F_\mu$. 
Acknowledgments

This work was supported in part by NSF award 0753845.

References


Received February 25, 2008; accepted July 18, 2008.