

A Temporally Asymmetric Hebbian Network for Sequential Working Memory

Jared C. Sylvester (jsylvest@umd.edu)

James A. Reggia (reggia@cs.umd.edu)

Department of Computer Science, A.V. Williams Bldg., College Park, MD 20742 USA

Scott A. Weems (sweems@casl.umd.edu)

Michael Bunting (mbunting@casl.umd.edu)

Center for Advanced Study of Language, 7005 52nd Avenue, College Park, MD 20742

Abstract

Recurrent connections combined with the appropriate dynamics enable oscillatory neural networks to produce rhythmic activity patterns. Such oscillatory activity can represent multiple stored patterns simultaneously, rather than the single pattern of a fixed-point network. However, retrieving these stored patterns in the same order as they were seen has proven challenging. In this paper we modify a recently developed simple oscillatory memory capable of storing temporal sequences so that it will now retrieve remembered items in the same order presented. This was achieved through the use of a temporally asymmetric weight matrix. The network is still capable of matching the recall performance of human subjects, reproducing the recency effect they exhibit in working memory tasks and displaying similar position-specific recall rates. We conclude that augmenting simple oscillatory neural network models with temporally asymmetric synaptic connections substantially improves their ability to match human short term memory properties.

Keywords: neural network models; autoassociative memory; short-term working memory; Hebbian learning; serial order

Introduction

There has been increasing interest in recent years in the development of oscillatory neural network models for a variety of tasks. In contrast to fixed-point attractor networks, which are typically limited to activating a single pattern in memory at a time, oscillating networks have dynamics characterized by recurrent connections leading to persistent rhythmic activity. This allows multiple patterns to be held in the same short-term memory concurrently as the model's state persistently switches between them.

A large variety of oscillating neural models exist. For example, some are based on underlying theta/gamma activity in the hippocampus or neocortex (Hasselmo, Bodelon, & Wyble, 2002; Ingber, 1995; Lisman & Idiart, 1995), while others use individual spiking neurons (Raffone & Wolters, 2001). Other more abstract approaches have also been used, for example Wilson-Cowan oscillators (Chakravarthy & Ghosh, 1996; Wang, 1995).

Here we concentrate on modeling short-term working memory, which is active over periods of time on the order of several seconds. A key characteristic of working memory is that it has a very limited capacity, unlike long-term memory (Baddeley, 2000). Recent studies suggest that this capacity is capped at around four items (Cowan, 2001; Cowan et al., 2005). More specifically we concern ourselves with modeling working memory for sequential tasks, or those for which the serial order of stimuli is important.

There is ongoing debate within cognitive psychology about the proper model of serial memory. Leading theories include the chaining model, ordinal theory, and positional theory (Henson, 1999). Recently focus has moved to connectionist neural network-based models (Brown, Preece, & Hulme, 2000; Burgess & Hitch, 1999). Here we present an approach that is reminiscent of the chaining model but avoids some of its drawbacks (see Discussion).

An elegant and parsimonious approach to oscillating working memory models is based on simple modification of Hebbian associative memories with fixed-point attractors to make them oscillatory. For example, Horn, D., Usher, M. (1991) developed a simple oscillatory memory by adding "dynamic thresholds" into Hopfield networks. With this approach, the thresholds used to determine the next activity state of a node are continuously changing such that it becomes increasingly difficult for a node to remain in the same state, and eventually it switches its activity state to the complementary value. When such a network is presented with multiple input stimuli it is found to oscillate between activity states representing these stored memory patterns.

We recently extended the Horn and Usher model to include a weight decay term so that the order of input pattern presentations could affect the network's recall (Winder, Reggia, Weems, & Bunting, 2009). This allows the network to accurately model the recency effect observed in human working memory on running memory span tasks. Stimuli which were presented later in the input sequence were more likely to be successfully stored and recalled by the network when using weight decay.

While the previous version of our model was able to match the position-specific recall rates of human subjects, the order in which the stimuli were recalled by the model was arbitrary. In this paper, we extend our oscillatory weight decay network to enable it to recall inputs in the order presented. The approach is to introduce a second set of temporally asymmetric weights into the model. By doing so we hypothesized that the network would be induced to oscillate between stored memory states in the desired order.

More specifically, we introduce into our simple oscillatory networks for the first time the use of temporally asymmetric Hebbian learning. Adaptation occurs in a fashion inspired by experimental evidence that synaptic efficacy in biological cortex and other brain structures is "temporally asymmetric" (Bi & Poo, 2001; Markram, Lubke, Frotscher, & Sakmann, 1997;

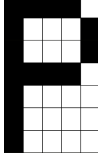


Figure 1: Stimuli to the model consist of 35 binary-valued inputs, conceived of as letters (such as the ‘P’ shown here) for ease of visualization and interpretation.

Zhang, Tao, Holt, Harris, & Poo, 1998). That is, synapses are strengthened (LTP) if presynaptic activity precedes excitatory post-synaptic potentials by 20-50ms, and weakened (LTD) if the time course is reversed. Our model, when extended in this fashion, not only captures the recency effect of the original model but also now largely retains the sequential order in which the stimuli were presented.

Methods

Model Description

Our model uses a fully connected network of N linear threshold units. Each node takes a binary value $a_i \in \{-1, 1\}$. The stimuli used are in effect arbitrary sets of N bits, though we consider them as being individual letters from A to Z for ease of interpretation. Figure 1 shows an input to a 35 node network interpreted as the letter ‘P.’

The operation of the model occurs in two phases: first a temporal sequence of input stimuli are presented and the weight matrices learned according to Eqs. 1 and 2 below, and then the model is allowed to oscillate between states according to Eqs. 3 and 4 for a predetermined total number of iterations. One iteration, or time step, corresponds to asynchronously updating every node once.

There are two sets of connection weights, W and V . Both are $N \times N$ matrices composed of real values, and are initialized to zero before learning. The first of these, W , is the same symmetric weight matrix used in previous version of this model (Winder et al., 2009). The entries of W are updated as each stimulus is presented according to:

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

where k_d is a decay rate ($0 \leq k_d < 1$), and δ_{ij} is Kronecker’s delta, which ensures that weights on self-connections are fixed at zero. This is, at its core, the same Hebbian weight change rule used in many previous neural network models. The difference is the addition of the decay term that reduces the influence of older stimuli in favor of more recent ones.

The new element of this model is the incorporation of a second weight matrix, V . The purpose of V is to allow the model to recall stimuli in the same order they were presented. In order to accomplish this, V is trained with a temporally asymmetric learning rule

$$v_{ij}^t = (1 - k_d)v_{ij}^{t-1} + \frac{1}{N}a_i^t a_j^{t-1} \quad (2)$$

inspired by the learning method used in some past neural networks for processing temporal sequences (Schulz & Reggia, 2004). This is similar to the Hebbian learning with decay given in Eq. 1, but it associates the activity of node i during the presentation of stimulus at time t with the activity of all other nodes j during the presentation of the *previous* stimulus at time $t - 1$ in the sequence. This introduces a sense of temporal ordering to the weight matrix, potentially making it possible to recall the stimuli in order rather than randomly as was previously done. Note that the decay term is still present, although the Kronecker’s delta factor is no longer used as it is desirable for a node’s activity to be influenced by its activation state in the previous time steps.

After learning and before recall the network is initially set in a random activity state. It is not necessary to prime the network with a partial or noisy version of any of the input patterns. The calculation of inputs to each node is modified from the prior model to account for both sets of weights. The input to node i at time step t is given as

$$h_i^t = \sum_j \left(\beta_1 w_{ij} a_j^t + \beta_2 v_{ij} a_j^{t-1} \right) - \theta_i^t \quad (3)$$

where the constant coefficients β_1 and β_2 are used to weight the relative contributions of W and V ($0 \leq \beta_1, \beta_2 \leq 1$). As in the previous version of the model, θ_i is a dynamic threshold used to insure that the network oscillates between states rather than coming to rest at a fixed attractor. Its calculation has been simplified from previously, however, with it now being updated according to the following two rules. Every time step, θ_i decays according to $\theta_i^{t+1} = (1 - k_\theta)\theta_i^t$. In any time step in which the state of node i has remained unchanged from the previous time step a factor of $k_w a_i^t$ is also added to θ_i^{t+1} . This moves θ_i in the direction of the activity state of node i , making it more difficult for node i to remain in the same state. Both k_θ and k_w are constants chosen in advance, with $0 < k_\theta < k_w < 1$. We use $k_\theta = 0.09$ and $k_w = 0.175$ in the following computational experiments, though similar values gave qualitatively similar results. Equation 3 has been simplified from the prior model by dropping the K_i biasing term derived from Horn, D., Usher, M. (1991). This was previously used to account for the potentially uneven distribution of active and inactive nodes across potential stimuli and current network state. Computational experiments revealed that it added computational complexity to the model without significant impact on performance.

After the input to each node is calculated, the node’s state is updated according to the following rule

$$a_i^t = \begin{cases} +1 & h_i^t > 0 \\ a_i^{t-1} & h_i^t = 0 \\ -1 & h_i^t < 0 \end{cases} \quad (4)$$

This is also a simplification of our earlier model, which used a stochastic updating process. We have found that the deterministic rule given above performs roughly the same with our data set and reduces computational cost.

Measuring Recall

We assess the network’s recall by calculating the Hamming distance d_λ between its activity state \vec{a} and \vec{a}^λ , where \vec{a}^λ is a perfect representation of one of the 26 stimuli λ :

$$d_\lambda = \frac{1}{2} \sum_{i=1}^N |a_i^\lambda - a_i| \quad (5)$$

The greater the distance d_λ between \vec{a} and \vec{a}^λ , the lower the similarity $s_\lambda = 0.85^{d_\lambda}$ will be. A value of $s_\lambda = 1.0$ indicates a perfect match between \vec{a} and \vec{a}^λ . We call any such time step a “recall peak” for λ . An exponential function was used to define s_λ in order to emphasize the difference between some pairs of inputs with small Hamming distance between them. The choice of 0.85 in the definition of s_λ is essentially arbitrary, chosen because it produced visually reasonable results. Values such as 0.7 or 0.9 work just as well.

In order to compare versions of the model as to whether they successfully recalled the stimuli in the same order as they were presented, we track the transitions from one recall peak to the next and use this to generate a single scalar value. We count the proportion of these peak-to-peak transitions which occur between one stimulus and the stimulus which was presented to the network immediately following. A transition from the fourth-back to the third-back stimulus would be counted as a correct transition, while one from the third to the fourth, or fourth to second, would not. A higher proportion of such correct transitions is indicative of the recall being more well ordered in the sense that the model is cycling through the stimuli it recalls in the same order as they were initially presented. Transitions following the one-back stimulus (i.e. the final stimulus) are ignored because there is no “next” stimulus to correctly transition to.

The recall phase of the model lasts for hundreds of time steps, each one potentially generating the recall of a stimulus. This lengthy series of activity must be distilled into a single ordering of the inputs, in which each unique stimulus appeared no more than once. This is accomplished by consolidating any consecutive time steps in which in the network peaks for the same stimuli. (Neither human subjects nor the model were ever presented with duplicates of the same stimulus, so there was no cause for the model to report seeing the same stimulus repeated.) So, for instance, if a stimuli sequence of “A B C D E” were to result in the network oscillating between the states “B C C C D D E” then the recalled sequence would be taken to be “B C D E,” and the second through fifth stimuli would be considered to have been remembered correctly. The requirement to remember the stimuli in the appropriate position is the same as what human subjects are faced with when doing running memory span tasks. Previous versions of the model were not subjected to this requirement; any recall peak for a stimulus was enough for it to be considered correctly stored.

Human Behavioral Data

We used behavioral data that we collected previously (Winder et al., 2009) 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 was obtained from 38 adult subjects 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. 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 (e.g., 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 backwards 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 which 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 which was 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). A total of twelve trials were conducted for the task with each subject requiring roughly 20 minutes per trial; no time restrictions were placed on subject responses. All 38 subjects completed the task.

Results

In the previous version of this model (Winder et al., 2009), the network was given an advantage in that it did not have to recall stimuli in the correct temporal sequence for them to be counted as correctly stored. Any network activity pattern during testing with sufficient similarity to an input was considered successfully stored, no matter when that activity pattern occurred. Here we increase the difficulty of the task by requiring the network to also recall stimuli in the correct sequence.

Figure 2 shows an example of the effect that introducing asymmetric weights has on sequential recall. A plot of peaks in similarity for each of the stimuli presented is shown. In Figure 2a without temporally asymmetric weights the ordering of the peaks is largely random, with the network moving between the four stored memory states without regard to their original presentation order. In contrast, Figure 2b with asymmetric weights shows that recalled memory patterns are much more ordered in their progression, with activity tending to proceed from earlier to later input patterns. This ordered retrieval of stored memories is much closer to the human behavioral task described above than was our earlier model.

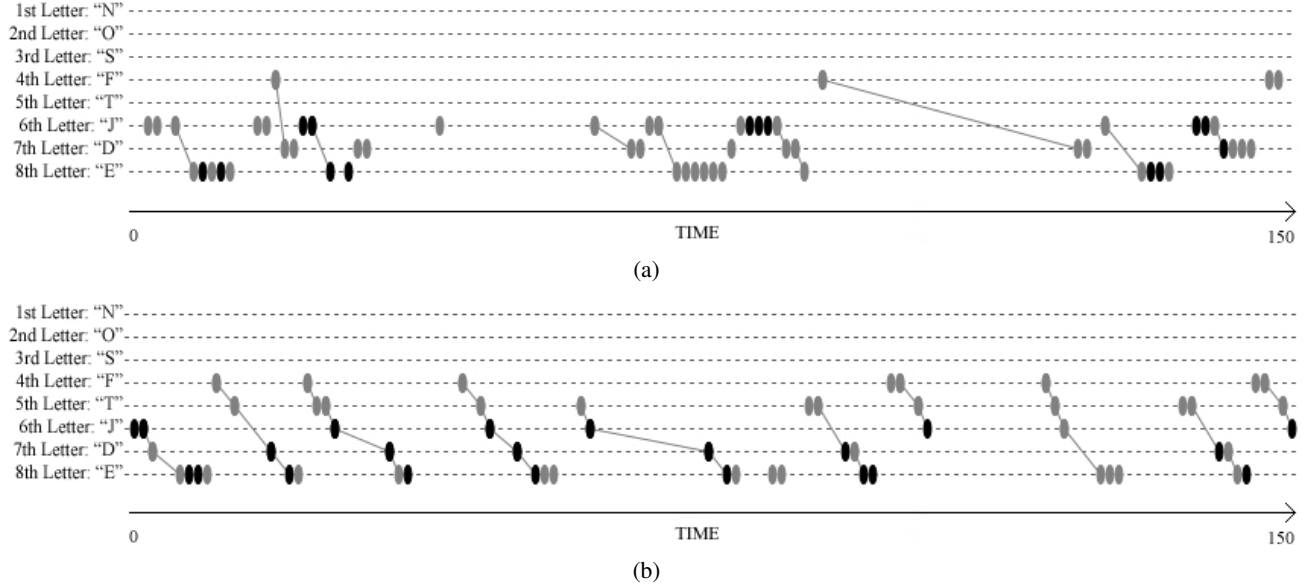


Figure 2: Plot over time of when the values of s reached their peaks for the eight stimuli during an example run of the model. Black marks indicate when s reached the maximum possible value of 1.0 and thus were counted as present, while gray marks indicate when s exceeded 0.8 but did not reach 1.0. The lines between activity peaks indicate transitions that occurred in the same order as the stimuli were presented. The first 150 time steps of the recall phase are shown here. Figure 2(a) is without asymmetric weights ($\beta_1 = 1.0, \beta_2 = 0.0$), and Figure 2(b) is with asymmetric weights ($\beta_1 = 0.5, \beta_2 = 1.0$). In the former, one can see that the oscillatory states alternate between the four recalled memory patterns for the 4th, 6th, 7th and 8th stimuli (F, J, D and E). Note that these peaks largely occur in an arbitrary order. In the latter case, the network state alternates between the five most recent stimuli, i.e. it has a propensity to recall input stimuli in the same sequence as that in which they were presented.

Table 1: Number of stimuli recalled.

		β_2				
		0.0	0.25	0.5	0.75	1.0
β_1	0.00	–	1.13	1.38	1.46	1.54
	0.25	1.18	1.84	2.01	2.22	2.12
	0.50	1.44	1.91	1.89	2.04	2.26
	0.75	1.72	1.88	1.95	2.02	2.08
	1.00	1.76	1.90	1.93	1.93	1.85

Table 1 shows the number of stimuli successfully stored and recalled by the network for various values of β_1 and β_2 when the network is presented with a sequence of six inputs. In constructing Table 1, five hundred random sequences were used for each simulation, and the network was allowed to oscillate for 250 time steps, with $k_d = .15$. The cell corresponding to $\beta_1 = 1.0, \beta_2 = 0.0$ is equivalent to running the network without any influence from the asymmetric weights. The best results were achieved with $\beta_1 = 0.5, \beta_2 = 1.0$, which gave a capacity of 2.26 items and with $\beta_1 = 0.25, \beta_2 = 0.75$, which gave 2.22 items. For comparison, human subjects had a memory capacity of 2.73 items and our previous model had a capacity of 2.69 (Winder et al., 2009). Note, however, that in the latter case the model’s recall was not required to be in the same temporal order as the stimulus.

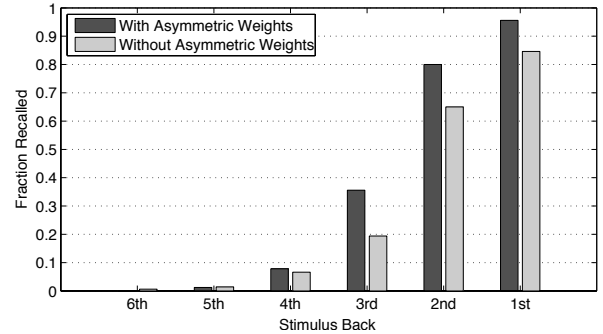


Figure 3: Recall rates for each position with and without temporally asymmetric weights. Five hundred random stimuli sequences were run using a decay rate of $k_d = 0.2$. Networks with asymmetric weights enabled used $\beta_1 = 0.5, \beta_2 = 1.0$.

In addition to increasing the total memory capacity relative to baseline ($\beta_2 = 0$), asymmetric weights also increase correct position-specific recall of the network. Figure 3 shows the recall rate at each stimulus position for networks both with and without asymmetric weights. Asymmetrically weighted networks were significantly more likely to retain the three most recent inputs.

Figure 4 shows that the network is capable of modeling

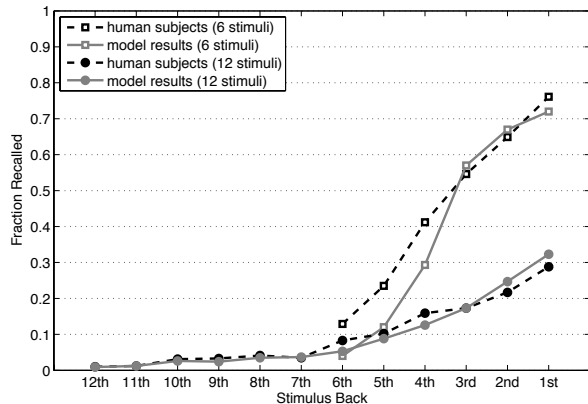


Figure 4: Comparison of the position-specific fraction of recalled stimuli by the model and human subjects for both 6-back and 12-back tasks.

human recency behavior on running span tasks when using asymmetric weighting by properly tuning the decay parameter, β_1 and β_2 . The model provides close matches for human performance on both 6-back and 12-back running span tasks. (For the former $k_d = 0.05$, $\beta_1 = 0.5$ and $\beta_2 = 1.0$ and for the latter $k_d = 0.075$, $\beta_1 = 0.63$ and $\beta_2 = 0.37$) Fitting data derived from human subjects is a simple matter of tuning these three coefficients, which was accomplished here with a simple iteratively-refined grid search, minimizing the RMSE.

In addition to having higher total and position-specific capacity, asynchronous weighted networks also retained the ordering of the input sequence more effectively. Table 2 gives the proportion of peaks in similarity s that occur in the correct order, using the same parameters as Table 1. That is, those that progress from the fifth-back to the fourth-back, for example. A high proportion of such transitions is achieved when the synchronous weights are ignored completely (i.e., when $\beta_1 = 0$), but note that the number of stimuli recalled by such networks is significantly lower (Table 1). The fewer items stored at all, the easier it becomes to get them into the correct sequence. Limiting the results to those networks which stored more than two of the six stimuli on average, we again find that $\beta_1 = 0.5$, $\beta_2 = 1.0$ gives the best result with 85% of the peaks in s transitioning correctly, compared to between 50 and 56% for the fully temporally symmetric networks, regardless of β_1 .

Discussion

This paper extends our earlier simple oscillatory memory model to bias it to produce ordered recall of input sequences. This extension maintains the intrinsic oscillatory nature of the previous model through the use of changing threshold values, and accounts for the ordering of input sequences with weights that include both associated (simultaneous) and temporally asymmetric components. The same results as the previous model, such as the re-creation of our human

Table 2: Portion of peak-to-peak transitions in correct order.

		β_2				
		0.0	0.25	0.5	0.75	1.0
β_1	0.00	–	.81	.86	.93	.87
	0.25	.56	.71	.71	.83	.78
	0.50	.50	.70	.68	.79	.85
	0.75	.56	.65	.68	.75	.78
	1.00	.53	.61	.67	.74	.71

subjects' recency effect, were maintained. The combination of weight decay and temporally asymmetric weight matrices allowed the model to do a much better job of recalling stimuli in the order they were originally seen while simultaneously boosting the number of stimuli successfully stored.

In addition, it was possible to match the model's performance to that from human subjects in two separate tasks (6-back and 12-back), specifically the existence of a prominent recency effect, by tuning only the decay rate and the balance between temporally symmetric and asymmetric influences. While our earlier model achieved success in matching the behavioral data, that model and human subjects were not being judged on the same scale, as the model did not have to recall stimuli in order while the human subjects did.

This model adds to the growing range of current models of short-term memory. It explains some of the richness of human memory behavior, for instance the recency effect in sequential recall tasks, but does so while remaining parsimonious in its design. There is no need in our model to explicitly specify lateral inhibition in order to provoke competition between stored patterns, such as in Haarman and Usher (2001). In contrast, competition is allowed to arise from the process of Hebbian learning and dynamic thresholds. Further, we do not use different structures for different phases of the memory process. There is no complex architecture of learning and recall units, or active gating structures to explicitly guide the recall process (Frank, Loughry, & O'Reilly, 2001; O'Reilly & Frank, 2006). Rather, a single substrate of identical nodes is all that is needed. The two weight matrices used in the model are also trained with nearly identical rules, and are treated identically during recall. There is also no need in our model to introduce extra layers or nodes to provide temporality of network activity, or to introduce recurrent connections or back-propagation between layers (Botvinick & Plaut, 2006). Multiple patterns, along with their order of appearance, can be stored on the same neural substrate simultaneously.

For the limited range of data considered here, our model did not need to maintain a unified record of the entire sequence of stimuli. Correlations between temporal events can be reconstructed by the network during recall in order to preserve the entire sequence, despite the network only being aware of the immediately preceding stimulus during training. The model's temporal "awareness," such as it is, only exists

in a thin temporal slice. Similarly, during the recall phase, each change of a node's activity is only dependent on the immediately preceding state of the network. Of course, with more complex data additional processing mechanisms would be needed.

While our model can be viewed as a variety of "chaining," it is important to recognize that it does not suffer from one of the principle weaknesses of chaining as a technique for storing sequences. Because of the inherently stochastic nature of the network's activity in the face of rapidly adjusting weight thresholds, there is little harm in being "knocked out" of sequence as the model is able to pick up the trail again. In fact, the initial state of the network is already out of the desired sequence: it is initialized to a random pattern, and not a noisy or partial version of the first pattern in the sequence like with many auto-associative networks. From this initially random state it is able to progress through the stimuli sequence, usually in the correct order, and only occasionally going astray but even then tending back towards the proper ordering. Note that other difficult conditions for chaining, such as duplicate stimuli and repetitions, were not present in the tasks that human subjects performed, and so were left out of our model's training as well.

An obvious direction for future research in this area is the introduction of additional sets of asymmetric weights. Just as we have one set of weights which refer back one time step into the past, it is possible to have a set of weights which refers to earlier activity, perhaps increasing effective sequencing of recall. Such an enhancement may help to deal with some of the difficulties of sequence learning mentioned above, such as repetitions, that were not addressed here. The previous version of the model also only needed a single parameter, k_d to be adjusted in order to match human behavioral data. By introducing β_1 and β_2 we have complicated this slightly. This could be ameliorated by using a single parameter to control the balance between symmetric and temporally asymmetric weights.

Acknowledgments: Supported, in whole or in part, with funding from the United States Government, including NSF award IIS-0753845.

References

- Baddeley, A. (2000). Short-term and working memory. In E. Tulving & F. Craik (Eds.), *The oxford handbook of memory*. Oxford Univ. Press.
- Bi, G., & Poo, M. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, *24*, 139-166.
- Botvinick, M., & Plaut, D. (2006). Short-term memory for serial order: A recurrent neural network model. *Psychological Review*, *113*(2), 201-233.
- Brown, G., Preece, T., & Hulme, C. (2000). Oscillator-based memory for serial order. *Psych. Rev.*, *107*(1), 127-181.
- Bunting, M., Cowan, N., & Saults, J. (2006). How does running span work? *Journal of Experimental Psychology: Human Perception and Performance*, *59*(10), 1691-1700.
- Burgess, N., & Hitch, G. (1999). Memory for serial order: A network model of the phonological loop and its timing. *Psychological Review*, *106*(3), 551-581.
- Chakravarthy, S., & Ghosh, J. (1996). A complex-valued associative memory for storing patterns as oscillatory states. *Biological Cybernetics*, *75*, 229-238.
- Cowan, N. (2001). The magical number 4 in short-term memory. *Behavioral and Brain Sciences*, *24*, 87-185.
- Cowan, N., Elliot, E., Saults, J., Morey, C., Mattox, S., Hismjatullina, A., et al. (2005). On the capacity of attention. *Cognitive Psychology*, *51*, 42-100.
- Frank, M., Loughry, B., & O'Reilly, R. (2001). Interactions between frontal cortex and basal ganglia in working memory: A computational model. *Cognitive, Affective, and Behavioral Neuroscience*, *1*, 137-160.
- Haarman, H., & Usher, M. (2001). Maintenance of semantic information in capacity-limited short-term memory. *Psychonomic Bulletin*, *8*(3), 568-578.
- Hasselmo, M., Bodelon, C., & Wyble, B. (2002). Proposed function for hippocampal theta rhythm. *Neural Comp.*, *14*, 793-817.
- Henson, R. N. A. (1999). Coding position in short-term memory. *Int'l Journal of Psychology*, *34*(5-6), 403-409.
- Horn, D., Usher, M. (1991). Parallel activation of memories in an oscillatory neural network. *Neural Comp.*, *3*, 31-43.
- Ingber, L. (1995). Statistical mechanics of neocortical interactions: Constraints on 40-hz models of short term memory based on persistent spiking. *Phys. Review E*, *52*, 4561-4563.
- Lisman, J., & Idiart, M. (1995). Storage of 7 ± 2 short-term memories in oscillatory subcycles. *Science*, *267*, 1512-6.
- Markram, H., Lubke, J., Frotscher, M., & Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, *275*, 213-215.
- O'Reilly, R., & Frank, M. (2006). Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia. *Neural Comp.*, *18*, 283-328.
- Pollack, I., Johnson, I., & Knaff, P. (1959). Running memory span. *Journal of Experimental Psychology*, *57*, 137-146.
- Raffone, A., & Wolters, G. (2001). A cortical mechanism for binding in visual working memory. *Journal of Cognitive Neuroscience*, *13*, 766-785.
- Schulz, R., & Reggia, J. (2004). Temporally asymmetric learning supports sequence processing in multi-winner self-organizing maps. *Neural Comp.*, *16*(3), 535-561.
- Wang, D. (1995). Emergent synchrony in locally coupled neural oscillators. *IEEE Trans. Neural Netw.*, *6*, 941-7.
- Winder, R., Reggia, J., Weems, S., & Bunting, M. (2009). An oscillatory hebbian network model of short-term memory. *Neural Comp.*, *21*, 741-761.
- Zhang, L., Tao, H., Holt, C., Harris, W., & Poo, M. (1998). A critical window for cooperation and competition among developing retinotectal synapses. *Nature*, *395*, 37-44.