1992-02

Working Memories for Storage and Recall of Arbitrary Temporal Sequences

https://hdl.handle.net/2144/2083

Boston University
WORKING MEMORIES FOR
STORAGE AND RECALL OF
ARBITRARY TEMPORAL SEQUENCES

Gary Bradski

February, 1992

Technical Report CAS/CNS–92–003
WORKING MEMORIES FOR STORAGE AND
RECALL OF ARBITRARY TEMPORAL
SEQUENCES

Gary Bradski, Gail A. Carpenter, and Stephen Grossberg

Center for Adaptive Systems and
Department of Cognitive and Neural Systems
Boston University
111 Cummington Street
Boston, MA 02215

Abstract

A working memory model is described that is capable of storing and recalling arbitrary temporal sequences of events, including repeated items. These memories encode the invariant temporal order of sequential events that may be presented at widely differing speeds, durations, and interstimulus intervals. This temporal order code is designed to enable all possible groupings of sequential events to be stably learned and remembered in real time, even as new events perturb the system.

1. INTRODUCTION

Working memory is a kind of short term memory (STM) where a temporally ordered sequence of events can be temporarily stored and performed, yet can be quickly erased by a distracting event, unlike long term memory (LTM). There exists a large experimental literature about working memory [1], as well as a variety of models.

The present class of dynamically defined working memory models, called STORE (Sustained Temporal Order REcurrent), as developed in [2], and [3], encode the temporal order of sequential events in activation levels within the memory, such that larger activations code for earlier items. The ratio between previously active items remains invariant as new inputs enter the memory even at widely differing speeds, durations, and interstimulus intervals. The temporal order code is thus designed to enable all possible groupings of sequential events to be stably learned and remembered in real time, because invariant activity ratios imply a learnable invariance in competitive learning vector directions. Thus, these working memories enable chunks (compressed, categorical, or unitized representations) of variable size to be encoded in LTM in a manner that is not destabilized by newly arriving inputs. The large cognitive database that can be explained by such models is summarized in [3], including stored primacy, recency, and bowed activity patterns (Figure 1).

This paper extends the basic model presented in [3] to allow for representation of repeated items in working memory. It also employs a decay term that provides for more control of
the form of the working memory patterns. A technical use for such working memories in 3-D visual object recognition is described in [2] and [3].

2. STORE MODEL The original STORE model consists of a two layered, input gated system where the bottom layer is a competitive system and the top layer tracks the bottom layer activations (figure 1a). Inputs are presented in an arbitrary sequential order with varying inter- and intra-input durations allowed (figure 1b), but without repeats. Inputs get stored in STM with larger activations representing earlier entry into STORE. The top layer in STORE acts to support the bottom layer against undue competitive influence from new inputs. Equations for the this STORE system are as follows:

\[
\frac{dx_i}{dt} = [AI_i + y_i - x_i x - Bx_i]I
\]

\[
\frac{dy_i}{dt} = [x_i - y_i]I^c,
\]

where \(x = \sum_k x_k\), \(I \equiv \sum_k I_k\), \(I^c \equiv 1 - I\), and \(x_i(0) = y_i(0) = 0\). The input sequence \(I_i\) is given by \(I_i(t) = 1\) if \(t_i - \alpha_i < t < t_i\), else \(I_i(t) = 0\).

3. SHAPE OF STORED STM PATTERNS The network with \(B = 0\) was studied in [3]. It generated steep activation gradients in STM. For technical applications
and biological modeling, we need a way to control the shape of the activation gradients. The decay term $-Bx_i$ in (1) accomplishes this. It can be mathematically proved that larger values of $B$, $1 > B \geq 0$, yield more gradual primacy gradients by using the equations:

$$S_1 \equiv x(t_1) = 0.5(-B + \sqrt{B^2 + 4A})$$

$$S_i \equiv x(t_i) = 0.5(-B + \sqrt{B^2 + 4(A + S_{i-1})}), \quad i > 1,$$

and

$$x_i(t_i) = \frac{A}{x(t_i) + B}; \quad x_{i-1}(t_{i-1}) = \frac{x_{i-1}(t_{i-1})}{x(t_i) + B},$$

where $1 - B > A$. Figure 2 shows the results of a series of computer simulations in which the decay parameter $A$ is varied for different $B$ values. Inputs were presented one at a time from leftmost to rightmost. To achieve a uniform scale for comparison, all activations represented by the bar charts have been normalized by the total activity $x(t_7)$. The smoothing out of the STM gradient can be seen when $A = 0.09$ and $A = 0.01$ as $B$ increases from 0 to 0.7.
4. STORAGE OF REPEATED ITEMS Repeated items pose a problem for temporal order memories. The problem is particularly difficult for memories that encode order in state transitions, because repeated states are equivalent. For our approach, where order is encoded in the activation levels of distributed nodes, the problem is still difficult. We cannot just increase the activation level of a repeated item because that would destroy the order encoded by the relative activation between nodes. The solution proposed here is that STORE encodes items and their order while the input and output connections to and from STORE automatically create new item representations when an external event is repeated. As shown in figure 3, a preprocessor interacts with STORE via feedforward excitatory connections and feedback inhibitory connections to break up repeated items into spatially separate channels so that repeated items enter the STORE network as spatially separate inputs.

Figure 4 shows how the preprocessor uses randomly weighted input connections from each input channel into a winner-take-all competitive field. The field chooses its maximal input activation, and when the winning activity in the field exceeds a threshold, it excites a corresponding site in the STORE network. After the input turns off, inhibition from the top layer of the STORE network prevents further activation at the winning site in the winner-take-all network until the STORE network is reset. In this way repeated inputs always excite new sites in the winner-take-all network until all sites are exhausted. The network can be designed with an arbitrary finite number of storage sites.
Figure 4: A STORE model capable of encoding arbitrary lists in STM.

(a) Step-by-step response of STORE to: BABBCA

<table>
<thead>
<tr>
<th>Entry</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) Final response of STORE to other sequences:

For all runs, \( A = 0.01, \ B = 0.7 \)

Figure 5: Storage of lists with repeated events.
The equation for the winner-take-all network is:

\[ \dot{r}_j = -\alpha r_j + (\beta - r_j)(\gamma r^2_j + \delta I_i) - \gamma r_j \sum_{k \neq j} (r^2_k + \theta y_i) \]  

(6)

as in [4], where \( \alpha = 0.01, \beta = 2.0, \delta_i \) represent random connection strengths from the input \( I_i \) (\( I_i = 1.0 \) when on, \( 0.0 \) when off), \( \gamma = 10.0 \), and \( \theta = 50.0 \) is the large inhibitory feedback from the top layer of the STORE network.

Computer simulations of STORE’s response to sequences involving repeated items is shown in Figure 5. This figure displays as a bar chart the equilibrated activations of \( X \) nodes in the STORE network arranged as a linear array by node type. All \( A \) nodes are shown together, all \( B \) nodes shown together and so on. We used a network representing four items \( (A \rightarrow D) \) with seven nodes per item. Figure 5a shows the step-by-step entry of the sequence \( BABBCA \). Figure 5b, shows STORE’s final pattern in response to various other sequences. This simulation used the system of equations (1) and (2) with \( A = 0.01 \), and \( B = 0.7 \).

ACKNOWLEDGEMENTS G. Bradski is supported in part by DARPA (AFOSR 90-0083); G.A. Carpenter by British Petroleum (89-A-1204), DARPA (AFOSR 90-0083), Office of Naval Research (N00014-91-J-4100), and the National Science Foundation (NSF IRI 90-00530); and S. Grossberg by the Air Force Office of Scientific Research (AFOSR 90-0175), DARPA (AFOSR 90-0083), Office of Naval Research (N00014-91-J-4100), and the National Science Foundation (NSF IRI-90-24877).

References


