

Spatio–Temporal Memories for Machine Learning: A Long-Term Memory Organization

Janusz A. Starzyk, *Senior Member, IEEE*, and Haibo He, *Member, IEEE*

Abstract—Design of artificial neural structures capable of reliable and flexible long-term spatio–temporal memory is of paramount importance in machine intelligence. To this end, we propose a novel, biologically inspired, long-term memory (LTM) architecture. We intend to use it as a building block of a neuron-level architecture that is able to mimic natural intelligence through learning, anticipation, and goal-driven behavior. A mutual input enhancement and blocking structure is proposed, and its operation is discussed in detail. The paper focuses on a hierarchical memory organization, storage, recognition, and recall mechanisms. Simulation results of the proposed memory show its effectiveness, adaptability, and robustness. Accuracy of the proposed method is compared to other methods including Levenshtein distance method and a Markov chain.

Index Terms—Embodied intelligence, hierarchical structure, long-term memory (LTM), memory robustness, spatio–temporal memory.

I. INTRODUCTION

SPATIO–TEMPORAL memory is needed for processing sensory inputs, anticipation, learning, motor control, and goal formation. Therefore, developing models and architectures of spatio–temporal memory is essential for building embodied intelligent systems. Although different models and architectures were proposed in the literature, there is still no clear picture what neural network structures should be used for the spatio–temporal memory. This paper aims to study spatio–temporal memories for machine learning and proposes a novel architecture aimed at robust, tolerant, and flexible temporal sequence learning and recognition. It also presents a detailed analysis of such structures.

Typically, two types of memory dominate in neurobiological research of human brain—the short-term memory (STM) and the long-term memory (LTM). It is believed that they have different structural organizations and occupy different regions of the human brain [1]–[4]. The study of memory has a long history, and can be traced back to the famous “perseveration–consolidation hypothesis” of Muller and Pilzecker [5],

[6]. Since then, one of the pioneering works that influence research on memory structures is the “dual-trace theory” proposed by Hebb and Gerard in [7] and [8]. Basically, this theory suggested that the continuing neural activities of STM can lead to LTM. That is to say, the experiences are initially stored in STM, and through continuous neural consolidation, LTM can be created. Recently, many authors have revisited such theories, and have investigated the various aspects of the relationship between STM and LTM [9]–[13]. For instance, McGaugh presents a review of research into memory consolidation over the century since the original perseveration–consolidation proposal in [10]. Recent observations of neuronal activities in the basolateral amygdala (BLA) of cats through multiple microelectrode arrays provide compelling evidence in support of this hypothesis [11], [12]. Computational models of working memory were investigated in [13] in which, rather than separating the STM and the LTM for analysis, a common framework capable of modeling both STM and LTM and highlighting their interactions was proposed. The authors argued that by using a common framework, hierarchical usages of context signals may potentially enable modeling of STM and LTM at different levels, such as syllables, items, or chunks [13].

In this work, our focus is on the LTM cells structure and operation and their interactions with the STM. We aim to develop a biologically inspired model that has a number of properties useful in self-organizing, hierarchical memory. The LTM memory cells should be selective enough to distinguish different sequences, yet tolerate small distortions of the sequence order and composition.

Since most of human perceptions and mental processes (for example, natural language processing, planning, and decision making) are sequential, we build our understanding of the memory organization based on spatio–temporal sequence learning, anticipation, and prediction. A general review of the sequence learning for machine intelligence, such as inference, planning, reasoning, natural language processing, and others can be found in [14]. As far as the connectionist networks for the learning mechanisms of spatio–temporal sequence is concerned, Kremer provided a comprehensive review of the research development in this subject in [15].

Recurrent neural networks (RNNs) are among the most powerful and successful methods for developing sequential memory [16], [17]. Characterized by recurrent connections, RNNs are able to learn, retrieve, and predict sequential information based on past experience. There are some successful examples of RNNs used for sequence learning, including the sequential behavior learning of arm robots and robotic navigation systems [18], [19]. However, the traditional RNN network using backpropagation through time (BPTT) is unable to learn

Manuscript received December 04, 2007; revised December 12, 2008; accepted January 02, 2009. First published March 27, 2009; current version published May 01, 2009.

J. A. Starzyk is with the School of Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701 USA (e-mail: starzyk@bobcat.ent.ohiou.edu).

H. He is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: hhe@stevens.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2009.2012854

sequences with a long time lag [20]. This is because the back-propagation error signals vanish in the BPTT mechanism. To overcome this limitation, Hochreiter proposed a long short-term memory (LSTM) in [20]. The idea is to introduce a memory cell (constant error carousel) in an RNN to be able to sustain its activation for a long period of time. Recently, Jacobsson presented a nice review of the rule extraction for RNN related to finite state machines (FSMs) in [21], in which the author also discussed some important open research questions on this subject.

Associative memory (AM) methods have also shown some success for sequence learning and prediction. For instance, in [22] and [23], Wang investigated the usage of associative neural networks with delayed feedback connections for sequence learning and retrieval. These models use either heteroassociative memory networks (HANNs) [22] or multiassociative neural networks (MANNs) [23] to learn relationships between the primary input sequence with its pairing sequence simultaneously. Recent developments using AM for sequential memory include the dynamic heteroassociative memory [24], a combined model based on context layer and shift-register models [25], etc.

Wang *et al.* introduced several important concepts in sequence learning and prediction in their series of papers [26]–[29]: timing and chunking, anticipation, incremental learning, and hierarchical organization. For instance, in [26], a dual-neuron and a sequence-detecting neuron were introduced to make the system able to recall sequences with some common patterns. In [27], a hierarchical organization based on the chunking mechanism was proposed to improve the memory capacity. In [28], an anticipation mechanism was introduced for the system to match the current input with the predicted information, enabling one-shot learning in this model. In [29], the authors showed that the anticipation mechanism is also capable of incremental learning. With this series of models, Wang *et al.* showed that sequences can be effectively learned, retrieved, and predicted in a hierarchical organization. The study of a self-organizing neural network model operated in an anticipative fashion was developed in [30]. By using a time-delayed Hebbian learning mechanism, this model can learn and recall complex temporal sequence and was successfully applied to robot trajectory planning applications. This model also has nice robustness characteristic by using redundancy in the sequence representation.

Recent development in this area suggested that a hierarchical organization is critical for building spatio-temporal memory for learning, predicting, and adjusting motor actions. For instance, Hawkins, in [31]–[33], develops a “memory-prediction” theory based on his interpretation of the neocortex. This memory is called a “hierarchical temporal memory” (HTM) and the authors present four reasons to make the hierarchy a critical element of the model [31]: generalization and storage efficiency (shared representations), consistency with the spatial and temporal hierarchy of the real world, quick response, and covert attention. In [34], a hierarchical self-organizing map model for sequence learning was developed. This model includes two interconnected self-organizing maps with one on top of the other. By effective using of the context information, this model is able to classify and discriminate sequences efficiently with the presence of noise information.

Previously, we developed an anticipation-based temporal sequence learning model with hierarchical structure [35]. Our model is characterized by three important features: hierarchical organization, anticipation, and one-shot learning. In our model, a modified Hebbian learning mechanism is used to recognize sensory input sequences (for instance, of letters or numbers). In this hierarchical organization, winners-take-all (WTA) was used to select active neurons in lower hierarchical levels to provide the input to higher hierarchical levels. Anticipation is a key element for our model since it enables highly efficient one-shot learning. Interested readers can find a more detailed description of this memory model in [35].

Theoretical analysis and simulation results of our previous model [35] showed that this model can successfully learn and predict sequence information. However, there still exist some issues that need to be resolved for it to be useful in hierarchical temporal memory. The first issue is the exact-match problem. In the model of [35], for an output neuron to be the winner in the competition stage, all the input elements within a sequence must exactly match the stored sequence represented by that neuron. This is a strong condition and makes this memory ineffective when the input sequence is distorted, delayed, or not completely defined. For instance, if we have already learned the word “Mississippi” and we have the input sequence “Mississippi,” our model will consider this a new sequence (a new word) since it does not match exactly the learned sequence. As a consequence, one-shot learning will be executed to learn such new sequence. However, the human brain can easily identify that this word is very similar to the stored information “Mississippi,” and most likely we will consider this as a “known sequence” that does not require learning.

A similar problem is the partial sequence issue. For instance, the model in [35] requires the sequence to start at the beginning to exactly match the stored information. If a sequence is started from the middle of a word, for instance, “ssissippi,” it will be stored as a new sequence, rather than being recognized as a characteristic part of the stored sequence “Mississippi.” In this paper, we aim to solve this issue and provide a biologically inspired, robust, and adaptive memory architecture. Another form of error tolerance is related to the typoglycemia phenomena in recognition of text messages [36] where a specialized error tolerance is required. We need to make memory structures capable of such tolerances.

Human memory is robust to environmental changes, and has error tolerance capabilities useful for efficient resource utilization. The key to successful recall is familiarity with the sequence and its fit to the context in which the sequence is considered. In this paper, we try to integrate this capability into our proposed memory model. We expect that this kind of tolerance in sequential memory is also useful for building invariant representations of the perceived environment in embodied intelligence.

The final issue addressed in this work is memory activation decay. In the LTM model in [35], the output neuron strength is maintained at the same level throughout the entire learning and predicting procedure. On one side, this characteristic is desired in biological, intelligent systems to keep the strength of useful information in building associations, and guiding the action planning to achieve goals. On the other hand, it is also useful for stored information to decay as new knowledge be-

comes available. In this paper, we introduce a memory activation decay mechanism to integrate these two characteristics.

Thus, our effort in this paper is to further extend our previous model [35] to make it more functional, robust, and efficient by addressing the above issues. The rest of this paper is organized as follows. In Section II, we discuss a system level organization of the LTM with distributed spatio-temporal memory cells within sparse hierarchical structures and its overall architecture. This architecture is analyzed in detail in Section III, including its storage, recognition, recall mechanisms, and model dynamics analysis. In Section IV, we present detailed simulations and statistical analysis of the proposed model. In Section V, we discuss the error tolerance of the proposed model, and compare the recognition performance of the proposed model with those of the hidden Markov models (HMMs) and the Levenshtein distance method. Finally, a conclusion and brief discussion is given in Section VI.

II. SPATIO-TEMPORAL MEMORIES IN SPARSE HIERARCHICAL STRUCTURES

Spatio-temporal memories could be organized in a hierarchical sparsely connected structure of identical processing units that govern data-flow in sensory and motor pathways that implement embodied intelligence in robots or other devices. These identical processing units play a role of minicolumns observed in a human cortex [37].

Representation building and learning to act in accordance with a machine's own objectives is accomplished on each level of appropriate hierarchical spatio-temporal memories. These memories associate events and patterns that are observed by the machine through its sensors and build links to motor pathways responsible for control of machine's actions. Thus, the associations that a machine makes relate features observed simultaneously or in a short time window, to events that took place over long time periods and in different spatial locations. Associations involving a short time window are typical of image processing, pattern recognition, and classification, which for many years dominated the applications of artificial neural networks. However, once the observations to be associated are spread over a wide range of time windows, the problems of coding, sparse representation building, input recognition, and recall become much more difficult.

The range of problems to be solved in the memory structures include temporal sequence learning, STM and LTM, binding problems, sparse coding, stability control, attention control, sensory-motor coordination, goal creation, and evaluation of machine's actions, etc. We try to address some of these problems in this paper.

A. Overall LTM Architecture

Fig. 1 shows the organization scheme of the LTM. This organization includes two major parts. First, an early pattern matching is presented in the lower part of Fig. 1 (level 0). Such preprocessing may be responsible for feature selection and early representation building. The evidence of such preprocessing exists in V1, V2, and V4 layers of human visual cortex [38]–[40]. In neural network structures, preprocessing and pattern recognition can be accomplished by means of the modified Hebbian

learning described in [35]. Second major part of this memory organization is a hierarchically organized array of LTM cells (level 1 to level n). It contains a number of identical LTM cells organized in several levels of the memory hierarchy. The output of one hierarchical level becomes an input to the next level. Alternatively, the inputs to LTM cells can be obtained from the STM. Such inputs will be used to train and to activate LTM cells on a higher level. Thus, inputs to each level of the LTM may be obtained through either lower level processing or recalled from the STM. In addition, feedback links from a higher level to a lower level LTM are used to provide a mechanism for anticipation, attention, prediction, and motor control. We have included several feedback links in Fig. 1 and highlighted the winning neuron of each hierarchical level in Fig. 1. Event driven pointer signals are used to control recall of a sequence stored in a selected LTM cell.

Each level LTM cells are controlled by the end of sequence signal appropriate for this level of hierarchy. For instance, if level 1 LTMs store sequences of letters, space can be selected as the end of a sequence symbol, and if level 2 LTMs store sequences of words, period may be selected as the end of a sequence symbol. Detailed discussion of the neural network structure to generate the pointer signal, to predict elements of the input sequence, and to activate the sequence learning mechanism was given in [35]. Identical mechanisms can be used with this new proposed LTM cells.

The major focus of this research is the organization of complex sequence storage, learning, and retrieval in LTM cells. We start with a brief discussion of the pattern matching and representation building stage (level 0) with three hierarchical layers [35]. The neurons in the second layer are sparsely connected, at random, to a subset of neurons in the first layer. There is some overlap of the subsets of neurons in the first layer to which the groups of the second layer neurons project.

Local WTA mechanisms are used in the level 0 to obtain distributed representations of the sensory inputs. In a local WTA, a winning neuron from a specific neighborhood is selected. Thus, a distributed representation is obtained through activation of several local winners. Initially, all the weights w_i are randomly set to positive and negative values with their sum equal to 0. After a local winner is selected through competition at one hierarchical level, the weights of the winner (w_{mi} , where the subscript m denotes a winner and i is an input neuron that projects to the winner) are adjusted as follows. For the connections with the input neuron i active

$$w_{mi}(t+1) = w_{mi}(t) + \varepsilon \cdot (M(0) \cdot (1 - w_{mi}(t))). \quad (1)$$

For the connections with the input neuron i not active

$$w_{mi}(t+1) = w_{mi}(t) - \varepsilon \cdot (M(1) \cdot (1 - w_{mi}(t))) \quad (2)$$

where ε is a small adjustment, and $M(1)$ and $M(0)$ are the number of 1s and 0s the winner neuron received. In our current simulation, we set ε equal to 0.05. As sensory information becomes available, dynamically, each output neuron updates its activity, which enables an unsupervised learning.

This update of sensory information is handled in the event-driven mode. A new input event activates the pointer neuron to

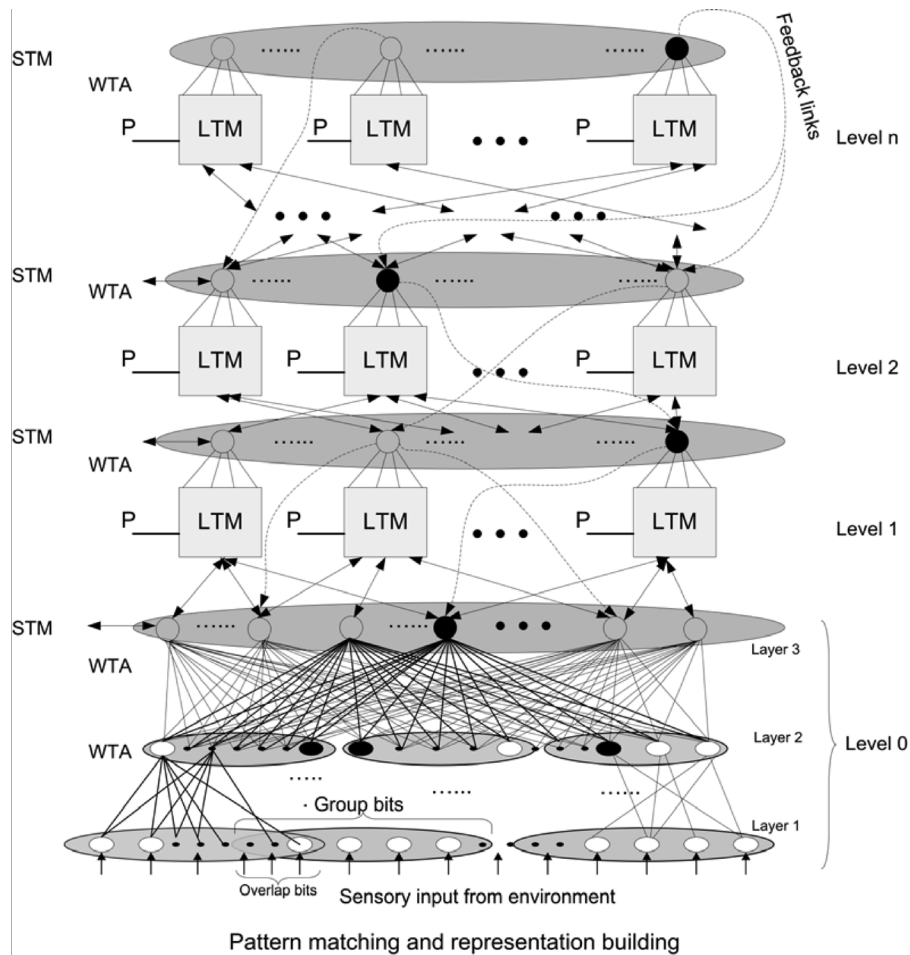


Fig. 1. Overall LTM architecture.

advance LTM cell processing. The sequence is ended with a special end of sequence signal different for different levels of distributed memory hierarchy. In a recall mode, the pointer neuron is activated by completion of the previous output control received from motor neurons. Chunks of lower level sequences (sentences or words) can be combined as sequences on a higher level and processed hierarchically as discussed in [35]. In a hierarchy of LTM cells, higher level sequences are processed based on completion signals from the lower levels. Thus, end of sequence signal on the lower level becomes an event that advances the pointer on the higher level.

In Fig. 1, the output of level 0 provides a symbolic representation for input sequences processed by level 1 LTM cells. Since artificial neural networks for sensory input representation building or recognition are well studied in many research papers, here we focus on organization and signal processing in LTM part of Fig. 1.

Hierarchy of sequences that are learned and processed by distributed memories composed of LTM cells in Fig. 1 is useful for machine learning. For instance, one can consider the input at level 1 to the LTM box as representing individual letters. These letters can either be recognized by the modified Hebbian learning structure in level 0 in Fig. 1 based on presented sensory input, or recalled from the STM on level 1. Subsequently, individual words can be learned as sequences of letters by means of the LTM cells and a WTA mechanism on level 1. At this par-

ticular hierarchical level, these words become LTM outputs that can be stored in STM on level 2. On a higher hierarchical level of the LTM memory (level 2), individual words become inputs to sentences learned through sequential combination. The process is repeated when one combines sentences into paragraphs. We will present the detailed LTM architecture, storage mechanism, recognition mechanism, and simulation analysis in Section IV.

Such distributed memories composed of cells that are capable to learn and recognize sequences are an important element of self-organizing systems that implement embodied intelligence. An example of such self-organizing system is presented in [41] where decision whether to learn a specific input representation is controlled by dynamically evolving goal creation system that governs machine development. Importance to develop contextual representation and sequence learning for developing language skills, comprehension, planning, complex motor control, temporally delayed reinforcement learning, and classical conditioning is broadly discussed in [42].

III. LTM ORGANIZATION

Described in this section, LTM cells are designed to store spatio-temporal sequences. We will describe the storage operation, input recognition, and recall functions of LTM cells. The memory was developed with several objectives to make it a reliable element of a self-organizing hierarchy of neurons employed in machine learning. We wanted to design LTM cells that can

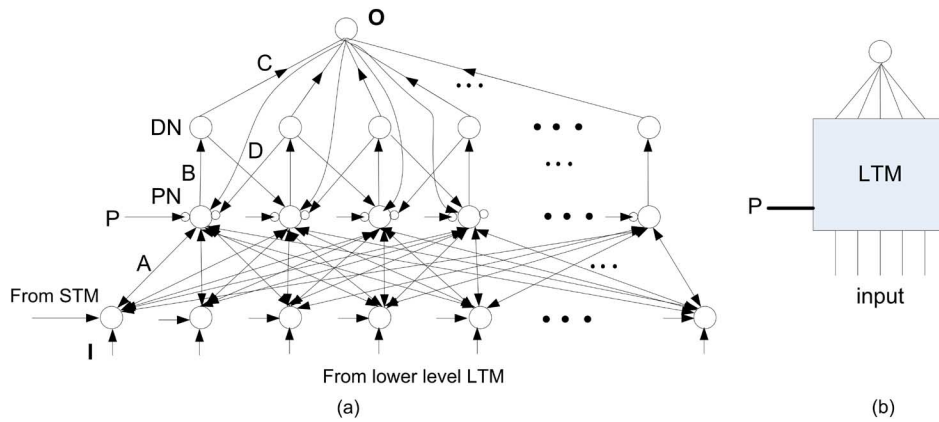


Fig. 2. (a) Details of LTM organization. (b) LTM cell symbol.

tolerate deviations from an ideal stored sequence and provide degree of the input sequence similarity to the stored sequence. We wanted the LTM cell to be able to respond to a matching sequence, even if the sequence is played from its middle terms, or is embedded in a sequence of different length than the stored sequence. This is particularly useful in sequence recognition task in such applications as continuous speech recognition when the sequence length varies with pronunciations. Finally, we wanted an LTM that performs its functions in parallel with other LTM cells using a sparsely connected structure that links the symbols it organizes in sequence.

In this section, we first describe the structural organization and the storage mechanism implemented by the LTM cells. This is followed by the description of recognition and recall mechanisms of such cells. Section III-A presents simulation results that test performance of the LTM cells. Statistical tests are performed to show the robustness of the proposed LTM mechanism. In the LTM discussed in this paper, we adopt the terminology introduced by Wang and Arbib in [26] and [27]. A temporal sequence is defined as

$$S : S_1 - S_2 \cdots - S_n \quad (3)$$

where S_i , $i = 1, \dots, n$, is a component of the sequence S and n is the length of S .

A. Storage Mechanism in LTM

Since typically an LTM will use an STM to provide it with data, we will describe, first, a mechanism for storing information in an LTM. To obtain learning data, either the existing LTM is searched for the relevant information and the search results are temporarily stored in the STM, or recent observations that are currently available in the STM or presented to the sensory input are used directly. The first option is used in learning sequences of events observed in the past, possibly at separate instants and on various time scales, while the second is used to learn most recent events that are represented in the STM or directly observed.

LTM cells compete for the best match during both storage and recall cycles. During storage, the input sequence is played exciting various LTM cells. The best match between the input sequence and the sequence stored in the LTM is determined

through competition and the winning LTM cell plays its sequence. If the match is judged not sufficiently accurate (below a specified threshold), a learning signal is generated for a new LTM storage. In embodied intelligence, this threshold will be learned through interaction with the environment and may be different in different areas of distributed memory, depending on the experience. Storage is performed through a competitive process. The winning cell adjusts its weights, storing the input sequence. All memory cells that were previously trained are excluded from the competition. This can be accomplished simply by using inhibitory links to the previously trained cells. The neuronal structures that implemented such selective allocation of memory cells in distributed self-organizing memories were discussed in [35]. Each LTM cell has the feedback structure shown in Fig. 2.

In this structure, a number of primary neurons (PNs) are linked to the outputs of STM or lower level LTM cells through bidirectional links A. PNs are also linked to inhibitory control signals from the sequence counter (pointer P). In untrained LTM, all links A are randomly set to small values. To store a new sequence, all untrained LTM cells compete and the best matching cell is declared a winner. Once an LTM cell is declared a winner in a learning competition, the links A between STM and PN as well as feedback links D are trained using one-shot learning. This learning takes place in a sequential fashion with the LTM neuron active during the learning cycle and only a single PN and a single-input neuron is active at any activation of the sequence pointer. Thus, for each subsequent element of the stored sequence, only a single feedforward link between one input and one PN is set to 1 and all other links to this PN are removed (weights are set to 0), and the corresponding feedback link D is set to 1.

All untrained feedback links D are inhibitory. All feedforward links B are fixed and set to 1. Untrained feedforward links C between dual neuron (DN) and the LTM neuron are mildly excitatory. Once the LTM neuron is activated during the learning cycle, all the links C between activated DN and LTM neurons are set to 1 and all other C links become inhibitory. Thus, if the input sequence is longer than the one stored in this LTM cell, inhibitory signals from untrained DN will lower the activation level of the LTM neuron, lowering its chance to win the competition for the best match.

The maximum sequence size that an LTM cell can store is determined by the number of PN and DN neurons it has. However, by chunking and using higher level hierarchies, arbitrary length memories can be obtained. Besides the A, B, C, and D links, each LTM cell has feedback links between consecutive elements of its sequence that are important for a successful recognition of the stored sequence (described next). Each new sequence stored in the LTM memory requires a dedicated LTM cell. Capacity of the LTM memory can increase if distributed representations are used. Such distributed representations for distributed spatio-temporal memory structures are currently under investigation.

B. Recognition Mechanism in LTM

An input sequence can be recognized by an LTM cell through activation of its PNs in the correct order. It is desired that a cell that stores a sequence identical to the one presented to the LTM will respond with the largest activation of its output neuron. A WTA mechanism is used to find the LTM cell with the best match.

A pseudocode of the recognition algorithm implemented by each LTM cell is presented next. In this algorithm, the following notation is used: $I(t)$ is the input vector excitation at time t , A is the set of activated PNs, β is DNs decay rate, $DN(t)$ are the dual-neuron activation values, $PN(t)$ are the primary-neuron activation values, W_I are the input weights, W_{DN} are the feedback weights from dual to PNs, $O(t)$ is the output activation, and O_s is the scaled output.

[Algorithm 1]: The LTM recognition algorithm

(1) Set $DN(0) = 0$, $t = 1$.

While not the end of the sequence do:

(2) Set $PN(t) = 0$, and change the input vector value $I(t)$.

(3) Find the set of activated PNs A .

(4) Partially discharge secondary neurons using a decay rate β

$$DN(t) = \frac{DN(t-1)}{\beta}, \quad \beta \geq 1. \quad (4)$$

(5) Update the activated PNs

$$PN_A(t) = I(t) W_I(A) + DN(t) W_{DN}(A) \quad (5)$$

where $W_{DN}(A)$ are columns of W_{DN} that correspond to activated PNs.

(6) Update the secondary neurons

$$DN(t) = PN(t), \quad (6)$$

(7) Update the output neuron activation

$$O(t) = O(t-1) + \text{sum}(DN_A(t)) - 1. \quad (7)$$

(8) Increment t by 1.

End loop

(9) Scale the output of the LTM cell as follows:

$$O_s(t) = \begin{cases} \frac{O(t)}{L_i \left(1 + \log_{10} \left(\frac{L_i}{L_s}\right)\right)}, & L_i > L_s \\ \frac{O(t)}{L_i \left(1 + \log_{10} \left(\frac{L_s}{L_i}\right)\right)}, & L_i < L_s \\ \frac{O(t)}{L_i}, & L_i = L_s \end{cases} \quad (8)$$

where L_i and L_s are the lengths of the input sequence and the stored sequence, respectively.

In this algorithm, the properties of LTM change with the value of β . When β is very large, the LTM changes to a set memory in which order of presentation is not critical, and lower values of β indicate strong sequential property of the LTM cell. In the LTM cell, a mechanism of mutual input enhancement and blocking (MIEB) is used to determine the input sequence order, and to allow discontinuous or out of order excitation. As we can see from Fig. 2, DNs are used to store the excitation level and to enhance or deplete excitation of other elements of the temporal sequence. The following examples illustrate the recognition mechanism in the LTM cell.

Example 1: Consider the LTM cell shown in Fig. 2 that stored the sequence “ABBA” and observe how this sequence can be recognized. As a result of learning, each PN is linked to a single-input neuron with the link weight set to 1. The weight of each link from a PN to its DN is 1. The excitation and inhibition feedback weights from the DN to the PN are 1 and 0.5, respectively. Thus, the weight matrix used in the algorithm presented in Algorithm 1 equals to

$$W_{DN} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.5 & 0 & 1 & 0 \\ 0 & -0.5 & 0 & 1 \\ 0 & 0 & -0.5 & 0 \end{bmatrix}. \quad (9)$$

By changing W_{DN} , various types of LTM cells are obtained with distinct tolerance to error. Such various types can be set by a designer, developed by using a genetic algorithm, or evolved through interaction with environment in self-organizing systems. At each iteration, the output neuron accumulates excitation from the activated DNs. The DN decay rate is $\beta = 2$. At each iteration, the output neuron’s excitation is decremented by 1 to offset subsequent inputs.

Table I shows the DN excitation level at each iteration step. At the final step (after the second A is presented), the output excitation level is normalized by the length of the sequence (four, in this case). In this way, the final output signal value is 1.3711.

It is useful to estimate the accumulated signal strength on the output of each LTM cell and use it as a normalization factor. The following theoretical analysis estimates the maximum signal strength in the proposed structures.

C. Model Dynamics Analysis

Assume that the excitatory feedback weights are equal to q^{n-1} , where n is the distance from a DN to its connected PN.

TABLE I
DUAL-NEURON EXCITATION LEVELS FOR THE CASE OF STORED SEQUENCE: ABBA AND INPUT SEQUENCE: ABBA

	DN1	DN2	DN3	DN4	Sum of the output of active DNs
Step1 - 'A'	1	0	0	1	1
Step 2 - 'B'	0	1.5	0.75	0	2.25
Step3 - 'B'	0	0.8125	1.75	0	3.8125
Step4 - 'A'	0.7969	0	0	1.875	5.4844

Proposition 1: The maximum output signal value in the LTM cell with DN decay rate $\beta = 1$ is determined by

$$\begin{aligned} S_{\text{out}} &= \sum_{i=1}^n s_i = n \left(1 - \frac{1}{q}\right) + \frac{1}{q} \sum_{i=0}^{n-1} (1+q)^i \\ &= n \left(1 - \frac{1}{q}\right) - \frac{1 - (1+q)^n}{q^2}. \end{aligned} \quad (10)$$

Proof: If there is no signal decay, then the maximum signal strength received at a DN location is equal to

$$s_i = 1 + s_{i-1}q^0 + s_{i-2}q^1 + \dots + s_1q^{i-2}, \quad i = 1, \dots, n \quad (11)$$

where $s_1 = 1$. So the maximum output signal value is determined by the series

$$S_{\text{out}} = \sum_{i=1}^n s_i = \sum_{i=1}^n \left(1 + \sum_{k=1}^i s_{i-k}q^{k-1}\right). \quad (12)$$

We can evaluate this series explicitly by expanding the geometrical series

$$\begin{aligned} s_1 &= 1 \\ s_2 &= 1 + s_1q^0 = (1+q)^0 + 1 = (1+q)^0 + s_1 \\ s_3 &= 1 + s_2q^0 + s_1q^1 = (1+q)^1 + 1 = (1+q)^1 + s_2 \end{aligned} \quad (13)$$

and, in general

$$\begin{aligned} s_i &= (1+q)^{i-2} + s_{i-1} = 1 + \sum_{k=1}^{i-1} (1+q)^{k-1} \\ &= 1 - \frac{1 - (1+q)^{i-1}}{q} \\ &= \left(1 - \frac{1}{q}\right) + \frac{(1+q)^{i-1}}{q} \end{aligned}$$

giving the maximum output signal strength as stated in Proposition 1.

Using a similar approach, the following proposition is true.

Proposition 2: When the signal at each dual node decays with the decay rate d , then the maximum output signal value is determined by the series

$$s_i = 1 + \frac{r[1 - r^{i-1}(1+q)^{i-1}]}{1 - r(1+q)} \quad (14)$$

where $r = 1/d$, and the final result is

$$S_{\text{out}} = \sum_{i=1}^n s_i = \frac{n(1-rq)}{1-r(1+q)} - \frac{r[1 - r^n(1+q)^n]}{[1 - r(1+q)]^2}. \quad (15)$$

Therefore, (15) provides the estimated signal value, and one can normalize the output sum according to this expected cumulative strength.

D. Recall Mechanism in LTM

Recall of the stored sequence from LTM is relatively easy. It can be initialized either through a bottom-up recognition of the sequence of representations at the lower level and a matching mechanism as discussed in the storage mechanism in LTM, or through a top-down link activating the LTM cell from a higher level representation (typically to perform a desired sequence of operations). In either case, a selected LTM memory neuron is activated. Subsequent elements of the stored sequence will be activated by the sequence pointer removing inhibition to a successive PN in the LTM cell and by a feedback signal from the LTM neuron to the PN through a trained link D (see Fig. 2). Notice that no activation of a PN will take place beyond the length of the stored sequence.

This top-down activation implements system anticipation if the LTM cell responds to signals in the sensory pathway or implements top-down control signal for the motor pathway [41]. The idea that brain is always matching expected results with the observed ones was discussed in [43]. It was also used in [42] as a foundation for the error driven learning in hierarchical structures of neurons simulating cortical columns. Spatial attention and anticipation was used there to solve the binding problem in object recognition when system perceives multiple objects.

IV. ANALYSIS OF LTM CELL PROPERTIES

A. Basic Mutual Input Enhancement and Blocking Simulation

The LTM cell architecture was developed to recognize a stored sequence even if the sequence was distorted or presented out of order. For instance, the LTM cell from Fig. 3 that stores sequence "ABBA" can recognize other sequences of these four letters; however, the recognition strength will be lower as indicated in Table II for different input sequences. When the presented input sequence is the same as the stored sequence, the output neuron will provide the highest activation signal. Otherwise, it will output a weaker activation signal.

Thus, the output activation signal normalized to the stored sequence activation level can be used as a similarity measure between two sequences. Similar sequences have the normalized excitation level close to 1, while dissimilar sequences similarity measure will be close to zero.

An extension of MIEB is mutual multiple input enhancement and blocking (MMIEB), where feedback links (excitatory or inhibitory) have diminishing strength for distant neurons. MMIEB with n -feedback links use weights of $1/2^{n-1}$, where n is the

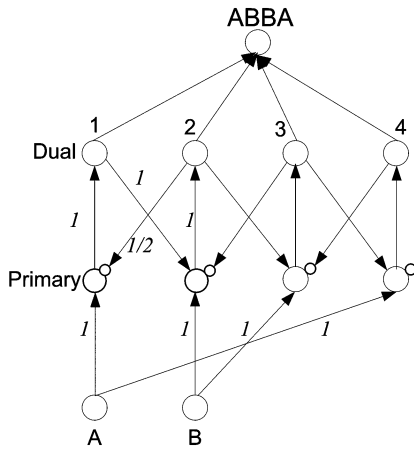


Fig. 3. Mutual input enhancement and blocking architecture.

TABLE II
NEURON EXCITATION LEVEL FOR DIFFERENT INPUT SEQUENCES
(ASSUME THAT THE STORED SEQUENCE IS "ABBA")

Input sequence	Output excitation level	Normalized excitation level
ABBA	1.3711	1.0000
BBAA	1.3438	0.9801
ABAB	1.1953	0.8718
AABB	1.2656	0.9231
BABA	1.1953	0.8718
BAAB	1.1875	0.8661

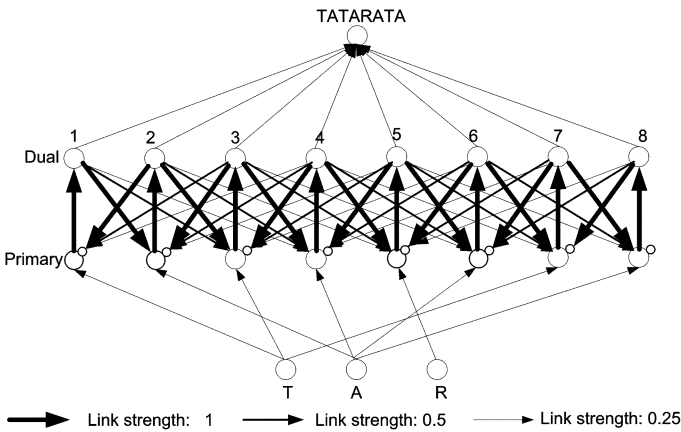


Fig. 4. Example of 3-MMIEB structure.

distance from the DN to its connected PN. The feedback structure in MMIEB cells is similar to the one used in higher order Markov chains [44], [45] with the exception that the connection weights are fixed and given. Fig. 4 shows the structure of the 3-MMIEB storing the sequence TATARATA.

Example 2: We simulated the MMIEB model as in Fig. 4 with three feedback links. The weights are 1, 0.5, and 0.25 for the nearest neighbor, next, and further away, respectively. We set the output decay rate to 2. In this simulation, we assume that the stored sequence is "perforation." The results were stored according to the normalized output signal strength in Table III. Here we divide the output strength by the value of the output for the stored sequence to obtain the normalized output strength column.

TABLE III
RESULTS FOR THE SAVED TEXT "PERFORATION"
WITH DECAY RATE EQUAL TO 2

Input text	Output strength	Normalized output	Input text	Output strength	Normalized output
perforation	1.9412	1.0000	prus	0.2171	0.1118
perforations	1.7147	0.8833	prast	0.4097	0.2111
peroration	1.4038	0.7232	barcelona	0.2363	0.1217
reformation	1.3070	0.6733	forest	0.4041	0.2082
defloration	1.2717	0.6551	preference	0.2922	0.1505
percolation	0.9765	0.5030	cat	0.1065	0.0549
performance	0.9929	0.5115	manifestation	0.1991	0.1026
deforestation	0.8806	0.4536	manifesto	0.1553	0.0800
penetration	0.6922	0.3566	aba	0	0.0000
perversion	0.6452	0.3324	bba	0	0.0000
prerogative	0.6031	0.3107	abb	0	0.0000
perk	0.4342	0.2237	abbba	0	0.0000
mantilla	0.1167	0.0601	abbaba	0	0.0000
gorilla	0.3527	0.1817	abba	0	0.0000

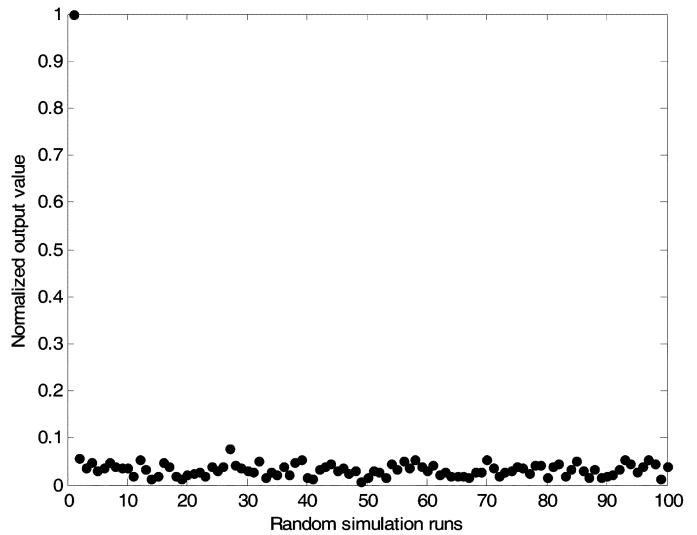


Fig. 5. Testing of the 3-MMIEB structure for the entire alphabet sequence.

Longer stored test sequences produce better, more robust separation between the output scores of the stored and permuted sequences. Similar to our analysis in Example 2, we use the stored alphabet from "a" to "z" (length 26) to test the performance of the MMIEB model. Here, we compose a sequence of the entire alphabet from "a" to "z" (length 26), and test the excitation level of the output neuron when different orders of this sequence are presented using the MMIEB model in Fig. 4 (DN decay rate equal to 2). To do so, we loaded the input sequence, and randomly permuted its order, and then calculated the output neuron excitation level according to Algorithm 1. All the neuron excitation levels were normalized to the original sequence activation level, plotted at location 1 in Fig. 5. Fig. 5 shows the simulation results for 100 random trials. As we can see, the stored input sequence is clearly separated from the distorted sequences.

Example 3: In this example, we simulate the general MMIEB model with n -feedback links use weights of $1/2^{n-1}$ (n is the distance from the DN to its connected PN) as presented in Fig. 4 for learning and recognition of regular English text. To do this, we use the original correct spelling of the typoglycemia test bench [36] as mentioned in Section I. In this test bench, the longest word is 13 letters. Therefore, we randomly flip k , $k = 1, \dots, 13$, letters and test the recognition results. To show

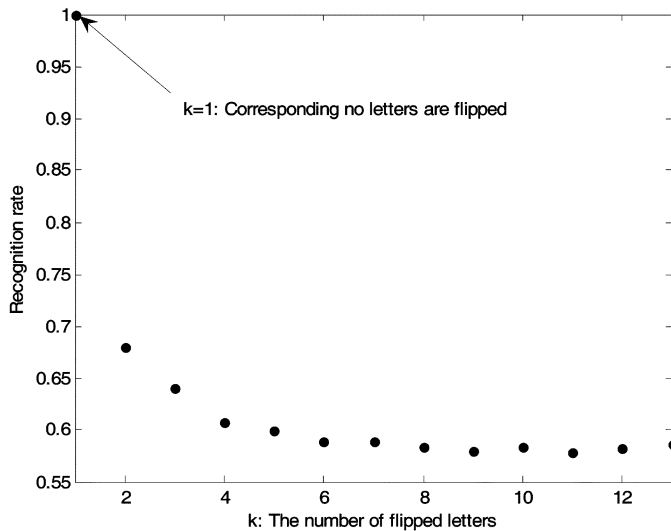


Fig. 6. Recognition of n -MMIEB model for the English text versus different number of flipped letters.

LTM performance with different decay rates

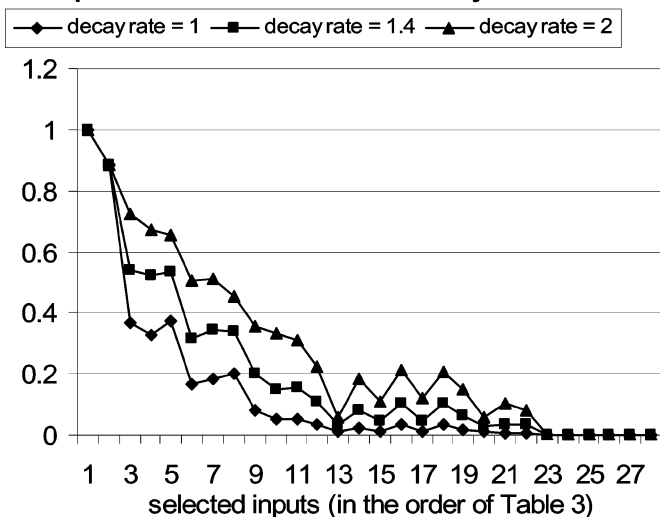


Fig. 7. Comparison of normalized output value for different decay rates.

the statistical performance, we conduct 100 random runs and Fig. 6 shows the average recognition rate versus the length of the flipped letters k . This result clearly indicates that the recognition rate of the MMIEB model decreases with the increase of the randomly flipped letters in the words.

B. Influence of Different Secondary Neuron Decay Rate

In order to test the influence of the secondary neurons' decay rate β , we repeated the simulation of Table III with different decay rates. Fig. 7 displays the results. In this figure, we show the normalized signal strength for different decay rates of secondary neurons. We can see that small secondary neuron decay rates seem to produce better separations. This can be characterized by estimated means and standard deviations of the permuted sequence matching value in comparison to the output value of the original sequence (normalized to 1). To obtain such

TABLE IV
MEAN AND STANDARD DEVIATION OF NORMALIZED OUTPUT VALUES OF RANDOM SEQUENCES OF DIFFERENT LENGTH

Decay rate	Sequence length	5	7	9	11	20
1	mean	0.009	0.009	0.007	0.005	0.0006
	std	0.018	0.014	0.010	0.006	0.0005
	Error percentage (%)	0	0	0	0	0
1.4	mean	0.014	0.015	0.015	0.016	0.010
	std	0.027	0.023	0.020	0.018	0.008
	Error percentage (%)	0	0	0	0	0
2	mean	0.018	0.024	0.030	0.037	0.053
	std	0.034	0.036	0.038	0.040	0.039
	Error percentage (%)	0	0	0	0	0

characterization, we statistically tested the LTM cell's performance for different decay rates. In this analysis, we use the general n -MMIEB model with n -feedback links and weights of $1/2^{n-1}$, where n is the distance from the DN to its connected PN (see Fig. 4 for an example of 3-MMIEB model). Rather than the sequence permutation analyzed in Fig. 5, here we generate random sequences of different lengths.

Example 4: We run 200 tests, and in each test we generate 100 random sequences for each selected length. The sequence recognition error probability is calculated from 20 000 random sequences of a given length. Table IV shows statistical results for various decay factors. As we can see from Table IV, we achieved a 100% correct recognition rate. Meanwhile, when the decay rate is reduced from 2 to 1, this method provides a much better separation margin.

As can be inferred from Proposition 2, the LTM cell signal strength increases rapidly with the length of the stored sequence. When it is desired to accumulate evidence of sequence matching, excessive signal strength may be a problem in hardware implementation of such memories. For this reason, LTM cells with a decay rate larger than 1 are useful, as the total signal strength in such cells is lower than when there is no decay (decay rate = 1).

V. TOLERANCE TO ERROR IN SEQUENTIAL MEMORIES

A. Need for Error Tolerance

Human sequential memories are error tolerant, as is apparent from the typoglycemia example in the introduction. Human memory recognizes familiar words by loosely matching their content (letters) guided by especially important clues such as the beginning and the end of a sequence [36], [51]. In this way, clues and vague recollections may activate past sequential memories, even if some parts of the sequence are missing or distorted.

A scheme that puts more emphasis on the first and the last element of the sequence can be coded using the MMIEB. We compare two implementations: one without special consideration for the beginning and the end of sequence and another one focusing on the first and the last element of a sequence, where these two elements are responsible for most of the feedback. Any element observed earlier than the correct beginning or later than the last element will introduce inhibition, lowering the total signal strength. In a similar way, we may stress the importance of the last element of a sequence by accumulating all the positive feedback received from other elements of the stored sequence.

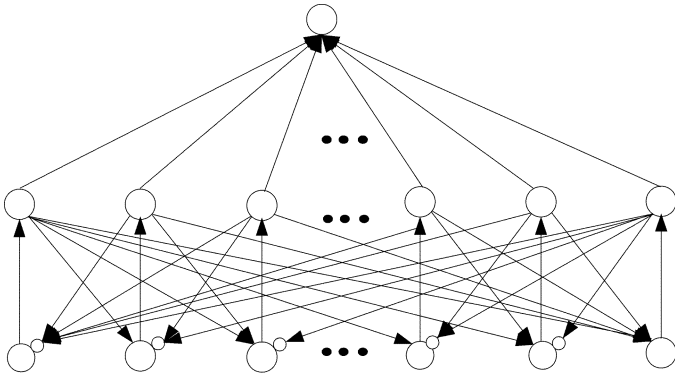


Fig. 8. Modified model with feedback focused on the first and the last element of the sequence.

Fig. 8 illustrates this architecture that stresses the importance of the first and the last element.

One should note that the structure in Fig. 8 can be considered as a special case of the general n -MMIEB model discussed in Section IV-A (Fig. 4). These structures differ in their feedback matrix W_{DN} from DNs to PNs (that can be learned if such training is provided).

Example 5: To verify the robustness and error tolerance of the modified model in Fig. 8, we test the model performance for the typoglycemia phenomena based on the following benchmark presented in [36].

“I cdn’uolt blveiee taht I cluod aulacly uesdnatnrd waht I was rdanieg: the phaonmneal pweor of the hmuan mnid. Aoccdrnig to a rseearch taem at Cmabrigde Uinervtisy, it deosn’t mtttaer in waht oredr the ltteers in a wrod are, the olny iprmoatnt tihng is taht the frist and lsat ltteer be in the rghit pclae. The rset can be a taotl mses and you can sitll raed it wouthit a porbelm. Tihis is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe. Such a cdonition is arppoiatprely cllaed Typoglycemia. Amzanig huh? Yaeh and you awlyas thguoht spleling was ipmorantt.”

We trained 73 LTM cells, each one for a unique word. Then, we loaded the misspelled words from the typoglycemia test bench, calculated the output value for each LTM cell, and determined the predicted word as the one with the strongest output value. Fig. 9 illustrates the simulation results. Stars represent the output signal strength when the structure of Fig. 8 is used, and circles represent the simulation results when a regular MMIEB structure as in Fig. 4 is used. From Fig. 9, we can clearly see that the modified structure (Fig. 8) can provide large output signal strength and better recognition in this situation, therefore it can provide better error tolerance. For the modified model, all these 107 words are correctly recognized, while for the regular MMIEB model, only 101 out of these 107 words are correctly recognized yielding the recognition rate of 94.39%. Table V shows the six words that cannot be recognized.

The proposed MMIEB model of LTM is flexible enough to accommodate various practical aspects of sequential memory such as those discussed in relation to various levels of tolerance. Optimization needed to obtain a desired level of performance

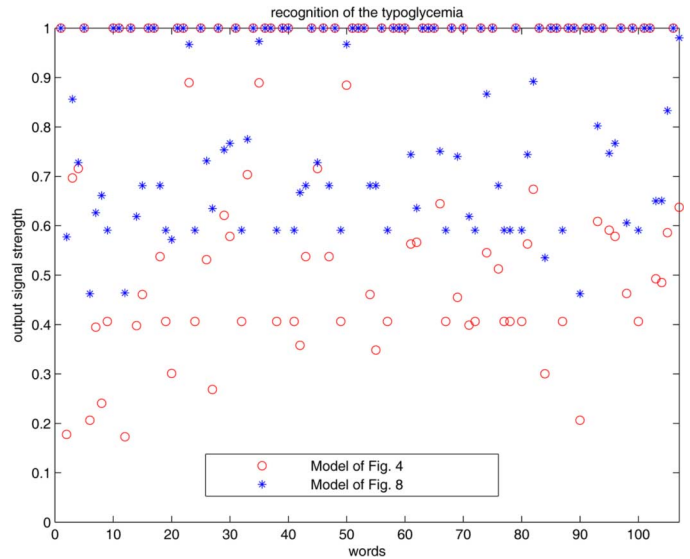


Fig. 9. Simulation of the typoglycemia phenomena.

TABLE V
UNRECOGNIZED WORDS BY THE REGULAR MMIEB MODEL
(TOTAL OF SIX WORDS)

Typoglycemia word	uesdnatnrd	rdanieg	lsat	tihis	deos	istlef
Corresponding correct word	understand	reading	last	this	does	itself
Misrecognized word by the MMIEB	phenomenal	believe	that	that	mess	letter

may result from interaction with the environment and various weighting schemes (passed genetically) can be adapted to various sensory inputs and learning mechanisms.

In a similar way, but to a lower degree, we may exhibit tolerance to the order of elements in playing back (recalling) the sequences from the memory. They are played back as original stored sequences as we write one letter at a time. Thus, even though, we may perceive a written word differently and with larger tolerance to error, we try to write it correctly, as each letter is written separately one by one. This is not to say that people perform this process without errors. But then they simply could be orthographic errors resulting from poor memory or dyslexic irregularities influenced by a learning disorder. This disparity between passive activity such as recognition of a sequence and its active equivalent (playing back the sequence) is easily accomplished in the proposed MIEB model.

B. Comparison of LTM Model With Other Methods

In this simulation, we test the recognition rates of the proposed LTM memory models with other methods including the HMM and Levenshtein distance method for the typoglycemia test bench from Section I. In this test bench, there are 107 words, among which we have 73 unique words. Therefore, for the HMM models, we train 73 HMMs, each for a unique word. Since each word has at most 26 distinct symbols (letters), we set the number of observation symbols to 26 in our simulation. The other parameters are set according to the suggestions in [46] for word recognition applications. That is to say, we set the number of states equal to 6, and the minimum discrete density value ε to 10^{-4} . The entire simulation is implemented in the Matlab environment with the toolbox of [47]. We train

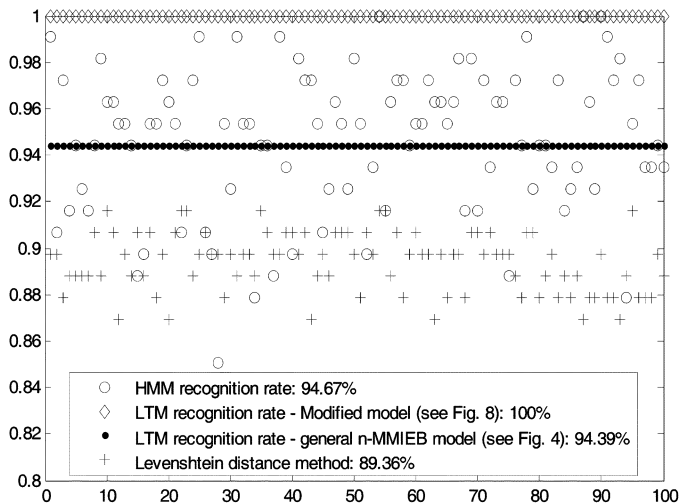


Fig. 10. Performance comparisons for sequence recognition.

the 73 HMM models with the corresponding correct words in the typoglycemia test bench, then we load the misspelled text as in typoglycemia to evaluate the log-likelihood of a trained model for each given test data, and predict the most likely word. The final recognition rate is calculated based on the ratio of correctly recognized words to the number of total words. For the Levenshtein distance method, we calculate the distance between each misspelled text as in typoglycemia with those of the correct words, and then predict the word as the one with the minimal Levenshtein distance. If a particular misspelled text under consideration has multiple potential correct words with the same minimal distances, we randomly select one of those words as the predicted sequence.

We run 100 random trials for all the methods. For the HMM model, we randomly set the initial condition for each run, and the average recognition rate is 94.67% with standard deviation of 0.0309. For Levenshtein distance method, the average recognition rate is 89.36%. As reported in Example 5, for the modified LTM model shown in Fig. 8, the recognition rate was 100%, while general MMIEB model had recognition rate of 94.39%. These results are illustrated in Fig. 10.

From these results, it seems that a regular MMIEB model can provide better recognition results compared to the Levenshtein distance method. However, the average performance of HMM is on a similar level. There are two observations that we can make. First, a regular MMIEB model performs equally well as HMM, but it does not require an extensive statistical training (needed to define an HMM model), since a one-shot learning step is used to store the input matrix W_I . Second, by training the feedback matrix W_{DN} , error tolerance to specific type of errors can be significantly improved over HMM model.

VI. CONCLUSION AND DISCUSSIONS

In this paper, we presented our research on spatio-temporal memory for machine learning with a focus on flexible, robust, and tolerant LTM organization. The key idea of this work is expressed through the LTM architecture and its memory cells based on mutual input enhancement and blocking. This

memory structure may serve as a basic building block in machine learning. We presented the detailed storage, recognition, and recall mechanisms of the LTM. Detailed simulation experiments and statistical analysis illustrate that the proposed memory organization is effective, robust, and fault tolerant. Finally, we compared the recognition rate of the proposed model with other models, and simulation results show the effectiveness of the proposed LTM model.

We demonstrated the application of this model to the text recognition application in this work. Specifically, we have tested its performance on the recognition of the typoglycemia benchmark and compared it to the HMM model. As a general spatio-temporal sequential memory organization, we believe that this model can also be generalized to many real-world applications that require complex sequence learning or sequential behaviors, such as speech recognition, natural language processing, video tracking, and others. The detailed application studies in such domains are out of the scope of this work, and will be reported in future research.

In the proposed model, a sequence (even if poorly presented and out of order) may be recognized correctly as long as it has enough characteristic features that distinguish it from other sequences. However, it will be played back without errors as a unique sequence exactly as it was stored, subject only to deterioration of memory or errors in the output translation. The output translation errors are related to errors in the mechanism that activates motor functions used to present this sequence. For instance, output translation takes place when we write a word on a piece of paper, when we pronounce the word by speaking, when we spell it out, or when we describe its meaning by drawing. Each of these output translations may introduce various types of errors related both to the way our memory stores and plays back sequences of motor neuron activation.

It is our intention to merge such LTM cells with STMs to obtain a learning memory that will build long-term representations only for these sensory inputs and associated motor activities that are relevant to a machine's goals. Learning signals, triggered by the reward or punishment inputs, will activate recall of the recent sensory and associated motor sequences. These sequences will then be stored in the LTM using mechanisms described in this work. A general goal-creation-based self-organization that can benefit from this kind of memories was presented in [41].

A major premise of such systems is that goal creation must result from machine's interaction with its environment. Therefore, an intelligent machine must have a built-in goal creation system (GCS) to create goals for its behavior. The main role of GCS is to develop sensory-motor coordination, goal-oriented learning of perceptions and actions, and to stimulate its interaction with the environment. The machine's memory is based on a distributed, hierarchical, self-organizing structure. The structure grows in complexity as goal hierarchy evolves. Meanwhile, the goal creation stimulates the growth of the heterohierarchy representing sensory inputs and a similar heterohierarchy representing actions and skills. The need for such systems is well expressed in [48].

The LTM model presented in this work is a simplification of a biologically feasible temporal memory process implemented by groups of neurons organized in minicolumns as suggested

in [49]. Using different computing resources, it will be possible to implement the proposed model in large associative networks that use sparsely coded representations. For instance, R-nets presented in [50] permit such associative recall of sparsely coded representations. In such a case, repetition of training cycles will yield a natural learning of sequences without the need of external control signals. Vogel [50] describes a simple AM mechanism for serial learning based on this principle. These kinds of memory models are needed if one aims at developing cognitive approach to design intelligent machines.

REFERENCES

- [1] B. Rypma, V. Prabhakaran, J. Desmond, G. H. Glover, and J. D. Gabrieli, "Load-dependent roles of frontal brain regions in the maintenance of working memory," *Neuroimage*, vol. 9, pp. 216–226, 1999.
- [2] J. J. Todd and R. Marois, "Capacity limit of visual short-term memory in human posterior parietal cortex," *Nature*, vol. 428, pp. 751–754, 2004.
- [3] D. Talmi, C. L. Grady, Y. Goshen-Gottstein, and M. Moscovitch, "Neuroimaging the serial position curve—A test of single-store versus dual-store models," *Psychol. Sci.*, vol. 16, no. 9, pp. 716–723, 2005.
- [4] L. G. Ungerleider, S. M. Courtney, and J. V. Haxby, "A neural system for human visual working memory," *Proc. Nat. Acad. Sci.*, vol. 95, pp. 883–890, 1998.
- [5] G. E. Muller and A. Pilzecker, "A. Experimentelle Beitrage zur Lehre vom Gedächtniss," *Z Psychol.*, vol. 1, pp. 1–288, 1900.
- [6] H. A. Lechner, L. R. Squire, and J. H. Byrne, "100 years of consolidation—remembering Müller and Pilzecker," *Learn. Memory*, vol. 6, no. 2, pp. 77–87, 1999.
- [7] R. W. Gerard, "Physiology and psychiatry," *Amer. J. Psychiatry*, vol. 106, pp. 161–173, 1949.
- [8] D. O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.
- [9] K. Touzani, S. V. Puthanveetil, and E. R. Kandel, "Consolidation of learning strategies during spatial working memory task requires protein synthesis in the prefrontal cortex," *Proc. Nat. Acad. Sci.*, vol. 104, no. 13, pp. 5632–5637, 2007.
- [10] J. L. McGaugh, "Memory—A century of consolidation," *Science*, vol. 287, pp. 248–251, 2000.
- [11] J. L. McGaugh, "Emotional arousal and enhanced amygdale activity: New evidence for the old perseveration-consolidation hypothesis," *Learn. Memory*, vol. 12, pp. 77–79, 2005.
- [12] J. G. Pelletier, E. Likhtik, M. Filali, and D. Pare, "Lasting increases in basolateral amygdala activity after emotional arousal: Implications for facilitated consolidation of emotional memories," *Learn. Memory*, vol. 12, pp. 96–102, 2005.
- [13] N. Burgess and G. Hitch, "Computational models of working memory: putting long-term memory into context," *Trends Cogn. Sciences.*, vol. 9, no. 11, pp. 535–541, 2005.
- [14] R. Sun and C. L. Giles, "Sequence learning: from recognition and prediction to sequential decision making," *IEEE Intell. Syst.*, vol. 16, no. 4, pp. 67–70, Jul.-Aug. 2001.
- [15] S. C. Kremer, "Spatiotemporal connectionist networks: A taxonomy and review," *Neural Comput.*, vol. 13, no. 2, pp. 249–306, 2001.
- [16] M. I. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," in *Proc. Conf. Cogn. Sci. Soc.*, 1986, pp. 531–546.
- [17] J. B. Pollack, "The induction of dynamical recognizers," *Mach. Learn.*, vol. 7, pp. 227–252, 1991.
- [18] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction process," *Neural Netw.*, vol. 16, pp. 11–23, 2003.
- [19] J. Tani and S. Nolfi, "Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems," *Neural Netw.*, vol. 12, pp. 1131–1141, 1999.
- [20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.
- [21] H. Jacobsson, "Rule extraction from recurrent neural networks: A taxonomy and review," *Neural Comput.*, vol. 17, no. 6, pp. 1223–1263, 2005.
- [22] L. Wang, "Learning and retrieving spatio-temporal sequences with any static associative neural network," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 6, pp. 729–739, Jun. 1998.
- [23] L. Wang, "Multi-associative neural networks and their applications to learning and retrieving complex spatio-temporal sequences," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 29, no. 1, pp. 73–82, Feb. 1999.
- [24] S. Chartier and M. Boukadoum, "A sequential dynamic heteroassociative memory for multistep pattern recognition and one-to-many association," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 59–68, Jan. 2006.
- [25] J. Bose, S. B. Furber, and J. L. Shapiro, "An associative memory for the on-line recognition and prediction of temporal sequences," in *Proc. Int. Joint Conf. Neural Netw.*, Montreal, QC, Canada, 2005, pp. 1223–1228.
- [26] D. Wang and M. A. Arbib, "Complex temporal sequence learning based on short-term memory," *Proc. IEEE*, vol. 78, no. 9, pp. 1536–1543, Sep. 1990.
- [27] D. Wang and M. A. Arbib, "Timing and chunking in processing temporal order," *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 4, pp. 993–1009, Jul./Aug. 1993.
- [28] D. Wang and B. Yuwono, "Anticipation-based temporal pattern generation," *IEEE Trans. Syst. Man Cybern.*, vol. 25, no. 4, pp. 615–628, Apr. 1995.
- [29] D. Wang and B. Yuwono, "Incremental learning of complex temporal patterns," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1465–1481, Nov. 1996.
- [30] A. F. R. Araujo and G. A. Barreto, "Context in temporal sequence processing: A self-organizing approach and its application to robotics," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 45–57, Jan. 2002.
- [31] J. Hawkins and D. George, "Hierarchical temporal memory-concepts, theory, and terminology," Numenta, Inc., Menlo Park, CA [Online]. Available: <http://www.numenta.com/>
- [32] J. Hawkins and S. Blakeslee, *On Intelligence*. New York: Times Books, 2004.
- [33] J. Hawkins and S. Blakeslee, "Why can't a computer be more like a brain," *IEEE Spectrum*, vol. 44, no. 4, pp. 20–26, Apr. 2007.
- [34] O. A. S. Carpinteiro, "A hierarchical self-organizing map model for sequence recognition," *Pattern Anal. Appl.*, vol. 3, no. 3, pp. 279–287, 2000.
- [35] J. A. Starzyk and H. He, "Anticipation-Based temporal sequences learning in hierarchical structure," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 344–358, Mar. 2007.
- [36] G. E. Rawlinson, "The significance of letter position in word recognition," Ph.D. dissertation, Psychology Dept., Univ. Nottingham, Nottingham, U.K., 1976.
- [37] V. B. Mountcastle, *Perceptual Neuroscience: The Cerebral Cortex*. Cambridge, MA: Harvard Univ. Press, 1998.
- [38] Y. Cao and S. Grossberg, "A laminar cortical model of stereopsis and 3D surface perception: Closure and da Vinci stereopsis," *Spatial Vis.*, vol. 18, pp. 515–578, 2005.
- [39] S. Grossberg, "How does the cerebral cortex work? Learning, attention and grouping by the laminar circuits of visual cortex," *Spatial Vis.*, vol. 12, pp. 163–185, 1999.
- [40] S. Grossberg and P. D. Howe, "A laminar cortical model of stereopsis and three-dimensional surface perception," *Vis. Res.*, vol. 43, no. 7, pp. 801–29, 2003.
- [41] J. A. Starzyk, "Motivation in embodied intelligence," in *Robotics, Automation and Control*. Vienna, Austria: I-Tech Education and Publishing, 2008.
- [42] R. C. O'Reilly and Y. Munakata, *Computational Explorations in Cognitive Neuroscience Understanding the Mind by Simulating the Brain*. Cambridge, MA: MIT Press, 2000.
- [43] J. L. McClelland, "The interaction of nature and nurture in development: A parallel distributed processing perspective," in *Current Advances in Psychological Science: Ongoing Research*, P. Bertelson, P. Eelen, and G. D'Ydewalle, Eds. Hillsdale, NJ: Erlbaum, 1994, pp. 57–88.
- [44] W. K. Ching, E. S. Fung, and M. K. Ng, "Higher-order Markov chain models for categorical data sequences," *Naval Res. Logistics*, vol. 51, no. 4, pp. 557–574, 2004.
- [45] S. Derrode, C. Carincotte, and S. Bourennane, "Unsupervised image segmentation based on high-order hidden Markov chains," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2004, vol. 5, pp. V-769–V-772.
- [46] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [47] Hidden Markov Model (HMM) Toolbox for Matlab [Online]. Available: <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>

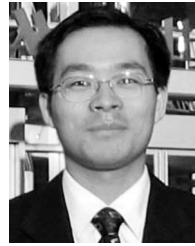
- [48] R. Tadeusiewicz and M. R. Ogiela, "Why automatic understanding?" in *Adaptive and Natural Computing Algorithms, Lecture Notes on Computer Science*, B. Beliczynski, A. Dzielinski, M. Iwanowski, and B. Riberiro, Eds. New York: Springer-Verlag, 2007, vol. 4432, pt. II, pp. 477–491.
- [49] J. Lücke and C. v. d. Malsburg, "Rapid processing and unsupervised learning in a model of the cortical macrocolumn," *Neural Comput.*, vol. 16, pp. 501–33, 2004.
- [50] D. Vogel, "A neural network model of memory and higher cognitive functions," *Int. J. Psychophysiol.*, vol. 55, pp. 3–21, 2005.
- [51] R. N. A. Henson, "Short-Term memory for serial order: The start-end model," *Cogn. Psychol.*, vol. 36, pp. 73–137, 1998.



Janusz A. Starzyk (SM'83) received the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from Warsaw University of Technology, Warsaw, Poland, in 1971 and 1976, respectively, and the habilitation degree in electrical engineering from The Silesian University of Technology, Gliwice, Poland, in 2008.

From 1977 to 1981, he was an Assistant Professor at the Institute of Electronics Fundamentals, Warsaw University of Technology, Warsaw, Poland. From 1981 to 1983, he was a Post-Doctorate Fellow and

Research Engineer at McMaster University, Hamilton, ON, Canada. In 1983, he joined the Department of Electrical and Computer Engineering, Ohio University, Athens, where he is currently a Professor of Electrical Engineering and Computer Science. He is an author or a coauthor of over 170 refereed journal and conference papers. His current research is in the areas of motivated embodied intelligence, goal creation in intelligent systems, sparse hierarchically organized spatio-temporal memories, self-organizing learning machines, and neural networks. He has cooperated with the National Institute of Standards and Technology. He has been a consultant to ATT Bell Laboratories, Sarnoff Research, Sverdrup Technology, Magnolia Broadband, and Magnetek Corporation. He was a Visiting Professor at University of Florence, Italy. He was a Visiting Researcher at Redstone Arsenal—U.S. Army Test, Measurement, and Diagnostic Activity and at Wright Labs—Advanced Systems Research and ATR Technology Development.



Haibo He (M'06) received the B.S. and M.S. degrees in electrical engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University, Athens, in 2006.

Currently, he is an Assistant Professor at the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ. His research interests include machine intelligence, self-adaptive systems, machine learning and data mining, computational intelligence and applications, very large scale integration (VLSI) and field-programmable gate array (FPGA) design, and embedded intelligent systems design.

Dr. He has served as a Program Committee member for several premium international conferences. He has also been a reviewer for the leading academic journals in his fields, including the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, and the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART B: CYBERNETICS, among others. He has delivered several invited talks including the IEEE North Jersey Section Systems, Man and Cybernetics invited talk on "Self-Adaptive Learning for Machine Intelligence." He was the recipient of the Outstanding Master Thesis Award of Hubei Province, China, in 2002. Currently, he is a committee member of the IEEE Systems, Man and Cybernetic Technical Committee on Computational Intelligence, and also an active member of the Association for Computing Machinery (ACM) and the Association for the Advancement of Artificial Intelligence (AAAI).