Neural Network Structure for Spatio-Temporal Long-Term Memory

Vu Anh Nguyen, *Student Member, IEEE*, Janusz A. Starzyk, *Senior Member, IEEE*, Wooi-Boon Goh, *Member, IEEE*, and Daniel Jachyra, *Member, IEEE*

***Abstract*— This paper proposes a neural network structure for spatio-temporal learning and recognition inspired by the long-term memory (LTM) model of the human cortex. Our structure is able to process real-valued and multidimensional sequences. This capability is attained by addressing three critical problems in sequential learning, namely the error tolerance, the significance of sequence elements and memory forgetting. We demonstrate the potential of the framework with a series of synthetic simulations and the Australian sign language (ASL) dataset. Results show that our LTM model is robust to different types of distortions. Second, our LTM model outperforms other sequential processing models in a classification task for the ASL dataset.**

***Index Terms*— Hand-sign language interpretation, long-term memory architecture, spatio-temporal neural networks.**

I. Introduction

THIS paper proposes a novel spatio-temporal long-term memory (LTM) architecture that is motivated by many neuro-biological evidences, such as hierarchical organization, fast learning, sparse connectivity, and error-tolerant retrieval [1]. We demonstrate that the proposed architecture significantly improves the original models in [2] and [3] in ways that make it more robust to process multidimensional and real-valued data. It is believed that the modeling of such memory is vital for the development of sensory-based representation and motor control of embodied intelligent systems.

Design of a sequential memory structure [4]–[7] has been known to mainly involve short-term memory (STM) and LTM. STM is used as a temporary storage of recent inputs for rapid processing and has a limited capacity [8]. Moreover, STM is able to store the order of the input events [9]. On the other hand, LTM is constructed using the synaptic modifications based on the consistent neural activities of STM [10] or the high synaptic plasticity in case of the episodic memory (EM) [11]. Our design of spatio-temporal memory architecture is based on the interplay between the two memory types.

Research on spatio-temporal neural networks dates back to the out-star avalanche model [12] and its variants [13]. The basic network consists of two layers, and has two key properties: temporal order invariance and normalization of activity. Time delay neural network [14] is a popular model that stores a sequence as a static multilayer feedforward network. Recurrent neural network (RNN) [15], [16] is another powerful family of sequential processing models. RNN introduces internal feedback links and a temporary buffer of recent states. The training of a typical spatio-temporal neural network is based on the back-propagation through time (BPTT) method [17]. However, the main problem with the BPTT method is that the gradient-based error signals may vanish or explode over a long lagging time [18], [19]. As a result, the training may fail to converge in practical time for problems with long-time dependencies. However, significant attempts [18], [20] were proposed to alleviate the difficulty of RNN's training.

Wang and Arbib [9] introduced several key issues related to complex sequence analysis, which include the temporal chunking, storage, retrieval, hierarchical organization, anticipation, and incremental learning. In [21], the same authors presented a model of sequence recognition by training the weight connections between STM and a sequence detector via a normalized Hebbian rule. More specifically, STM was modeled as an array of cells, each of which corresponds to an element of the input training sequence with decaying behavior. In the same paper, they also discussed distributed representation for complex sequence processing. In [22], the previous model was improved by adopting an online learning mechanism with anticipation.

Wang [23], [24] developed a general framework for complex sequence learning, storage, and retrieval applied to any hetero-associative/auto-associative neural network [23] or any multiassociative neural network [24]. His network's structure consists of three components: a voting network, an array of associative neural networks, and delayed feedback lines. Experiments with noisy alphabetical patterns showed that the framework is able to robustly learn and retrieve a large number of non-orthogonal patterns with good accuracy.

In our previous model [2], a spatio-temporal learning architecture that addresses several critical issues related

Manuscript received October 21, 2011; revised February 9, 2012; accepted March 11, 2012. Date of publication April 30, 2012; date of current version May 10, 2012.

V. A. Nguyen and W.-B. Goh are with the Center for Multimedia and Network Technology, School of Computer Engineering, Nanyang Technological University, 637553, Singapore (e-mail: anhngv102@gmail.com; aswbgoh@ntu.edu.sg).

J. A. Starzyk is with the School of Electrical Engineering and Computer Science, Russ College of Engineering and Technology, Ohio University, Athens, OH 45701 USA, and also with the Department of Applied Information Systems, University of Information Technology and Management, Rzeszow 35-603, Poland (e-mail: starzyk@bobcat.ent.ohiou.edu).

D. Jachyra is with the Department of Applied Information Systems, University of Information Technology and Management, Rzeszow 35-603, Poland (e-mail: djachyra@wsiz.rzeszow.pl).

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/TNNLS.2012.2191419

to LTM-based sequential memory structures, such as the hierarchical, sparsely connected organization, competition, anticipation, and one-shot learning was developed. In this context, an LTM cell provides a persistent storage of a sequence. At an architectural level, LTM [25] contains a number of identical processing units organized in hierarchical layers. The architecture of LTM resembles the organization of the cortical mini-columns of a human brain [1]. Hierarchical representation provides a natural way to tackle complex problems in which the higher layers exploit the lower layers to learn a large number of different patterns [26].

Similar to [22], the network proposed in [2] actively anticipates the next element with feedback connections. The learning of the input sequence is activated only when the anticipation of next elements is incorrect. However, the main difference to [22] is that the chunking of input sequences is done automatically, once the learning signal is triggered. The second advantage is the requirement of only a single presentation of a training sequence as opposed to [22], which requires multiple sweeps of the sequence. The model was able to learn any complex sequence as long as the number of distinct subsequences is smaller than the memory capacity provided by the neurons across the hierarchy.

Our model in [3] improved the work in [2] by introducing a flexible matching mechanism that gives a real-valued measure of similarity between the learnt and testing sequence instead of a discrete match-nonmatch score. It addresses the error tolerance problem of the neural network to a few types of uncertainties in a test sequence, including order distortion, time delay, and imperfect start–end segmentation of a sequence. This significantly improves the learning efficiency compared to the learning mechanism in [22], which triggers the one-shot learning whenever the test sequence does not exactly match the stored sequence. Comparison evaluation with existing methods, including hidden Markov model (HMM) and Levenberg–Marquardt algorithm on a task related to storage and prediction of words demonstrates its effectiveness in recognition accuracy.

This paper preserves the characteristics of the LTM models in [2] and [3], but extends their ability to perform robust recognition of real-valued and multidimensional sequences. Three main contributions are proposed: 1) the introduction of error tolerance within an LTM cell; 2) the incorporation of significance of elements in the LTM cell; and 3) the augmentation of the LTM framework with a novel activation decay mechanism. These contributions are briefly characterized next.

Error tolerance in sequential learning can be analyzed at inter-element and intra-element level. The inter-element type of error includes various distortions of temporal relationship among consecutive elements of the input. On the other hand, the intra-element error refers to various distortions in the content of the input. These two problems were not adequately addressed in [3] since the model assumed that each element of the sequence should be recognized perfectly, which means that it is either present or not. However, the problem is more complicated when sequences are continuously varying and multidimensional. In this paper, we introduce mechanisms to address these two error types, and they lead to substantial improvements in the recognition performance.

For intra-element error, we characterize the error tolerance of the content of each element by learning the statistical spatio-temporal variations. We show that the tolerance estimation of variation from only a single training sequence can be used to robustly recognize testing sequences. It is arguable that the tolerance of a sequence should be approximated from statistics of multiple samples of the target sequence in a probabilistic setting. However, in many situations, many training instances are not available and the system is expected to operate after a single observation. For example, a robot is required to learn a topological path after a single run through an environment [27]. Another example is a speech recognizer that learns a single presentation of a word spoken by a person, and is expected to remain partially tolerant to others speaking the same word. EM organization [11] that requires an agent to learn a sequence of events after a single observation can also benefit from our approach.

We address the inter-element error by a robust sequence recognition, which tolerates inter-element variability. When a testing sequence is presented, LTM cells incrementally accumulate evidences from the testing sequence and compete to be the winner. Theoretical analysis shows that only the stored sequence in the LTM cell elicits maximum activation and any deviation from the ideal sequence results in a graceful degradation of activation. The maximum activation of an LTM cell is analytically derived and used to normalize the activation. Thus, matching of the stored sequence with a test sequence of a different length is allowed.

Our second contribution deals with the significance of elements stored in each LTM cell. Due to the limited computational resource, an agent may choose to put more emphasis on identifying and processing only an important subset of elements. This complements a typical learning that assigns an unit significance to all the elements. The novelty of our model is the explicit modulation of the LTM activation by estimated elements' significance. We propose a specific significance analysis that is suitable for the chosen application based on the statistical variation of the sequence elements. We demonstrate that the incorporation of significance improves the recognition performance level.

It is well understood that the definitions and identification techniques of significant elements vary depending on specific applications. Three examples are motifs in DNA, RNA, and proteins sequences in bioinformatics [28], salient spatio-temporal events in dynamic scene understanding based on center–surround interactions [29] or statistical differences from subjective expectations [30], and sequences of events that are associated with predefined goals in robotic navigation [31]. In this paper, the significance of elements is integrated as a modulatory factor for *any* estimation method.

The last contribution of our network structure is the introduction of memory activation decay. The reasons for the activation decay are two-fold: 1) to maintain the strength of an LTM cell for a sufficient duration to perform learning, construct associations, and predict next events and 2) when the current sequence of events increasingly deviates from the

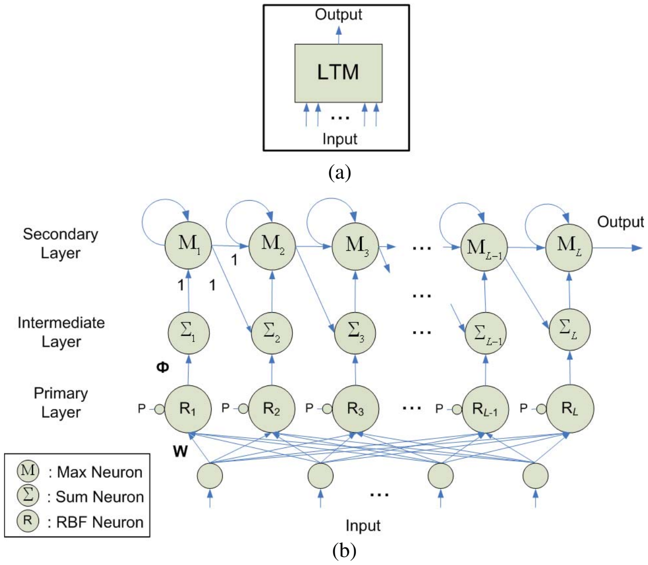


Fig. 1. (a) Block diagram and (b) detailed structure of an LTM cell.

LTM cell, the output strength needs to decay rapidly to avoid ambiguities in decision making. It will be empirically shown that by using memory activation decay, the separation of activations between a matched LTM cell and non-matched LTM cell given a test sequence increases, leading to an improvement of recognition performance. The issue of memory activation decay for sequential neural networks was discussed in several previous works [3], [11], [13], [21].

The structure of this paper is as follows. Section II presents the theoretical aspects of an LTM cell's structure and a novel recognition algorithm for real-valued multidimensional sequences. Section III discusses various properties of an LTM cells with synthesized sequences. Section IV presents the empirical results and comparisons of the LTM model with the Australian sign language (ASL) dataset [32]. Finally, Section V concludes this paper and discusses several future extensions to the framework.

II. LTM Cell Organization, Learning, and Recognition

Using notations similar to [2] and [21] a spatio-temporal sequence $\mathbf{S}$ is represented as $S_1 - S_2 - \cdots - S_L$, where $S_i (i = 1, \ldots, L)$ is a component of the Sequence, and L is the length of the sequence. Each of the component S_i is represented by a vector of features, i.e., $S_i \in \mathbb{R}^F$ where F is the dimension of the vector. In a matrix form, we have

$$\mathbf{S} \equiv \{s_{ij} | i = 1, \ldots, F, j = 1, \ldots, L\}. \tag{1}$$

A subsequence of $\mathbf{S}$ is any $S_m - S_{m+1} \cdots - S_n$, where $1 \leq m \leq n \leq L$. In addition, we denote the tolerance $\Delta \in \mathbb{R}^{F \times L}$ in a matrix form and the significance $\Phi \in \mathbb{R}^L$ in a vector form of the elements of the sequence as follows:

$$\Delta \equiv \{\delta_{ij} \in \mathbb{R}^+ | i = 1, \ldots, F, j = 1, \ldots, L\} \tag{2}$$

$$\Phi \equiv \{\phi_j \in (0, 1] | j = 1, \ldots, L\}. \tag{3}$$

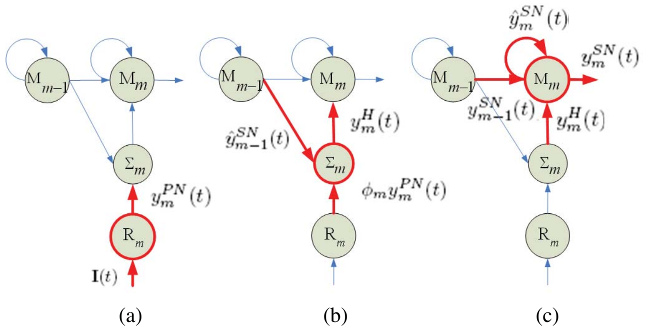


Fig. 2. Neuronal update process at the mth triple of an LTM at the time step t. See text for details. (a) PN's update. (b) IN's update. (c) SN's update.

The tolerance and significance of a sequence specify the statistical variation and the importance of the elements in the sequence, respectively. The previous works in [2] and [3] only dealt with a special case when $\delta_{ij} \approx 0, \forall i, j$ (i.e., either match or nonmatch) and $\phi_j = 1, \forall j$ (i.e., equal significance).

A. LTM Cell Structure

An LTM cell is designed as a building block to store an input sequence $\mathbf{S} \in \mathbb{R}^{F \times L}$ [see Fig. 1(a)], and subsequently determine if a given testing sequence is matched to the stored sequence by a graded signal. This paper proposes a sparsely connected LTM structure [see Fig. 1(b)] that is used as an efficient and robust basic component for a hierarchical sequential neural network. The network topology in Fig. 1(b) comprised of four layers. They are the input layer, the primary layer, the intermediate layer, and the secondary layer. The details of the network layers are described as follows.

1) *Input Layer*: The input layer consists of F input neurons. It provides the LTM cell with information obtained either from the sensory system that is connected to environment or the outputs from LTM cells at lower levels in a hierarchical organization. At a time step t, an input vector $\mathbf{I}(t) \equiv \{I_i(t) | i = 1, \ldots, F\}$ of a testing sequence is presented to the network.
2) *Primary Layer*: The primary layer consists of L primary neurons (PNs) depicted as the "R" neurons in Fig. 1(b). The content of a training sequence is stored as the synaptic weights $\mathbf{W} \equiv \{w_{ij} | i = 1, \ldots, F, j = 1, \ldots, L\}$ embedded within the full connections between the input layer and the primary layer. Each PN also has an inhibitory control signal from a sequence counter P, which is used for the learning of $\mathbf{W}$. The role of the primary layer is to compute the degree of similarity between an input vector and components of the stored sequence. The similarity can be computed by a pattern recognizer, such as a multilayer perceptron. In this paper, the radial basis function is employed as the similarity metric. Fig. 2(a) shows how the output (or *primary excitation*) of the mth PN ($m = 1, \ldots, L$) at the time step t (denoted as $y_m^{PN} \in [0, 1]$) is computed as

$$y_m^{PN}(t) = \exp\left[-\frac{1}{F}\sum_{i=1}^{F}\left(\frac{w_{im} - I_i(t)}{\delta_{im}}\right)^2\right] \tag{4}$$

where δ_{im} is the tolerance of the ith feature of the mth component defined in (2). The update for all the PNs is performed *concurrently*.

3) *Intermediate Layer*: The intermediate layer consists of L intermediate neurons (INs) depicted as the "Σ" neurons in Fig. 1(b). At the time step t, the mth neuron of this layer sums the outputs from the mth PN computed by (4) at the current time step, and the $(m-1)$th secondary neuron (SN) (described later) computed at the previous time step [denoted as $y_{m-1}^{SN}(t-1)$]. The one-to-one connection between the mth PN and the mth IN is weighted by the significance degree of the component, i.e., ϕ_m. Fig. 2(b) shows how the output of the jth IN at the time step t (denoted as $y_m^H(t)$) is computed as
$$y_m^H(t) = \left[\phi_m y_m^{PN}(t) + \hat{y}_{m-1}^{SN}(t)\right]^+ \tag{5}$$
where $\hat{y}_m^{SN}$ is the activation of the mth SN after being decayed and $[x]^+ = x$ if $x \geq 0$ and 0 otherwise. The neuronal decaying behavior of the SNs is described by the functions $f_j(.)(j = 1, \ldots, L)$ as depicted by the self feedback loop in the Fig. 1(a) and is given by
$$\hat{y}_j^{SN}(t) = f_j\left(y_j^{SN}(t-1)\right) \quad \forall j. \tag{6}$$
The basic requirement of the decaying function is
$$f_j(x) \leq x \quad \forall j \in [1, \ldots, L]. \tag{7}$$
The update for all the INs is performed *concurrently*.

4) *Secondary Layer*: The secondary layer consists of L SNs depicted as the "M" neurons in Fig. 1(b). Fig. 2(c) shows the output (or *secondary excitation*) of the SNs at the time step t (denoted as $y_m^{SN}(t)$) being updated *incrementally* by
$$y_m^{SN}(t) = \max\{\hat{y}_m^{SN}(t), y_m^H(t), y_{m-1}^{SN}(t)\} \tag{8}$$
where max{.} is the point-wise maximum function and $y_0^{SN}(t) = 0$ by convention. The updated activation of the mth SN provides a matching degree between the test sequence and the subsequence $S_1 - S_2 - \cdots S_m$ of the stored sequence. The activation of the mth SN is updated based on the maximum function of three different signals [in the order of the max function in (8)]. They are its decayed activation from the previous step, the newly updated mth IN after receiving the signal from the mth PN (which indicates the arrival of the mth component of the stored sequence), and the matching degree between the presented sequence and the subsequence $S_1 - \cdots - S_{m-1}$ of the stored sequence.

In this paper, we use the following linear functions for modeling the decaying behavior:
$$f_j(x) = x - \gamma\phi_{j+1} \ (\gamma \in (0, 1]). \tag{9}$$
Nonlinear decay for sequence recognition has been previously used in [13] to explain several psychological phenomena, such as *recency* (the elements near the end of the sequence are harder to forget than those at the beginning of the sequence) and *primacy* (the elements near the start of the sequence are harder to forget than those at the end of the sequence). Such phenomena require specialized knowledge of the remembered sequence that is not assumed here.

B. Model Dynamics

The output of the LTM cell at the time step t is given by the secondary excitation of the last SN, i.e., $y_L^{SN}(t)$. This activation provides a matching score between the input sequence presented until the current time step and the stored sequence in the LTM cell. This section provides proofs that the maximum activation of an LTM cell can only be attained by the stored sequence. This maximum activation can be derived analytically and be used for the LTM activation's normalization.

First, the following prepositions summarize the computation of the SNs.

Preposition 1: After each update, the activations of the SNs from 1 to L are monotonically increasing
$$0 \leq y_1^{SN}(t) \leq y_2^{SN}(t) \leq \cdots \leq y_L^{SN}(t). \tag{10}$$

Proof: This preposition is a direct result from the update rule (8). ■

Preposition 1 implies that after each update, the Lth SN always contains the maximum activation of all the SNs.

Preposition 2: At any time step t, the following property holds for any $m \in [1, L]$:
$$y_m^{SN}(t) = \max\{\hat{y}_m^{SN}(t), \max_{k \in [1,m]} [\hat{y}_{k-1}^{SN}(t) + \phi_k y_k^{PN}(t)]^+\}. \tag{11}$$

Proof: The following proof is applied at any time step t. Therefore, the time index is left out for compactness
$$\begin{aligned} y_m^{SN} &\overset{(8)}{=} \max\{\hat{y}_m^{SN}, y_m^H, y_{m-1}^{SN}\} \\ &\overset{(5)}{=} \max\{\hat{y}_m^{SN}, [\hat{y}_{m-1}^{SN} + \phi_m y_m^{PN}]^+, y_{m-1}^{SN}\}. \end{aligned}$$
Likewise
$$\begin{aligned} y_{m-1}^{SN} &= \max\{\hat{y}_{m-1}^{SN}, [\hat{y}_{m-2}^{SN} + \phi_{m-1} y_{m-1}^{PN}]^+, y_{m-2}^{SN}\} \\ &\ldots \\ y_1^{SN} &= \max\{\hat{y}_0^{SN}, [\hat{y}_0^{SN} + \phi_1 y_1^{PN}]^+, y_0^{SN}\} \end{aligned}$$
where $y_0^{SN} = 0$ by convention. By combining these equations, we have (11) for any $m \in [1, L]$. ■

Prepositions 1 and 2 are used to prove Lemma 1, which give the upper bound of activations of the SNs.

Lemma 1: Let $\beta_m = (1-\gamma)\sum_{j=1}^m \phi_j$ and given that the initial activations of the DNs satisfy the following conditions $y_m^{SN}(0) \leq \beta_m, \forall m \in [1, L]$, the activation of SN in subsequent steps satisfies the following inequality:
$$y_m^{SN}(t) \leq \beta_m \quad \forall m \in [1, L]. \tag{12}$$

Proof: We prove this lemma by induction.

1) $m = 1$: We have
$$\begin{aligned} y_1^{SN}(t) &\overset{(11)}{=} \max\{\hat{y}_1^{SN}(t), [\hat{y}_1^{SN}(t) + \phi_1 y_1^{PN}(t)]^+\} \\ &\overset{(9)}{=} \max\{\hat{y}_1^{SN}(t), [\phi_1 y_1^{PN}(t) - \gamma\phi_1]^+\} \\ &\overset{(4)}{\leq} \max\{\hat{y}_1^{SN}(t), (1-\gamma)\phi_1\} \\ &= \max\{\hat{y}_1^{SN}(t), \beta_1\}. \end{aligned}$$

From the condition of the initial value, we have $\hat{y}_1^{SN}(t) \overset{(9)}{=} f_1(y_1^{SN}(t-1)) \overset{(7)}{\leq} y_1^{SN}(t-1) \leq \beta_1$.

Therefore, (12) holds for $m = 1$. The equality happens when the first element of the LTM sequence is presented.

2) Assume (12) holds for all the SNs from index 1 to $m-1$ at a time step t

$$y_k^{SN}(t) \le \beta_k \qquad \forall k \in [1, m-1]. \tag{13}$$

The equality occurs when the first k elements of the stored sequence are presented in correct order. We need to prove that (12) also holds for the mth SN at the time step $(t+1)$

$$y_m^{SN}(t+1) \le \beta_m. \tag{14}$$

From the condition of the initial value, we have $y_m^{SN}(t) \le \beta_m$. Due to the decaying function, we have at the time step $(t+1)$

$$\hat{y}_m^{SN}(t+1) \overset{(9)}{=} f_m(y_m^{SN}(t)) \overset{(7)}{\le} y_m^{SN}(t) \le \beta_m. \tag{15}$$

The second component of the outer-max function in (11) takes into account the primary excitations produced by an input vector. Let

$$A(t) = \max_{k\in[1,m]} [\hat{y}_{k-1}^{SN}(t) + \phi_k y_k^{PN}(t)]^+$$

we have

$$\begin{aligned} A(t+1) &\overset{(4)}{\le} \max_{k\in[1,m]} [\hat{y}_{k-1}^{SN}(t+1) + \phi_k]^+ \\ &\overset{(9)}{=} \max_{k\in[1,m]} [y_{k-1}^{SN}(t) + (1-\gamma)\phi_k]^+ \qquad (16) \\ &\overset{(13)}{\le} \max_{k\in[1,m]} \{\beta_{k-1} + (1-\gamma)\phi_k\} \\ &= \max_{k\in[1,m]} \{\beta_k\} = (1-\gamma)\sum_{j=1}^{m} \phi_j = \beta_m. \qquad (17) \end{aligned}$$

From (11), (15), and (17) we have (14), which proves Lemma 1. The equality occurs when the first $(m-1)$ elements [by (17)] and the mth element [by (16)] of the stored sequence are presented in correct order. ■

From Lemma 1 and Preposition 1, we can derive the maximum activation of an LTM cell, $O_{\max}$, as follows:

$$O_{\max} = \max\{y_L^{SN}(t)\} = (1-\gamma)\sum_{j=1}^{L} \phi_j. \tag{18}$$

The maximum activation can be attained only by the stored sequence. This value is used to normalize the strength of the LTM cell's output during testing.

Algorithm 1 LTM Sequence Recognition (LTMSR)

```
Require: $\mathbf{W}, \Delta, \Phi, O_{\max}, \gamma$
Ensure $O$
Initialize:
  1) $y_m^{SN}(0) \leftarrow 0 \quad \forall m \in [1, L]$.
  2) $c_m \leftarrow \tau \quad \forall m \in [1, L]$.
  3) $t \leftarrow 1$.
Begin Algorithm:
  for each input vector $\mathbf{I}(t)$ of the test sequence do
    Compute $\{y_m^{PN}(t), \hat{y}_m^{SN}(t), y_m^{H}(t)\}, \forall m$ by (4)–(6).
    for $m = 1$ to $L$ do
      if $(\hat{y}_m^{SN}(t) \ge \max\{y_m^{H}(t), y_{m-1}^{SN}(t)\}) \wedge (c_m > 0)$ then
        $y_m^{SN}(t) \leftarrow \hat{y}_m^{SN}(t)$.
        $c_m \leftarrow c_m - 1$.
      else
        $y_m^{SN}(t) \leftarrow \max\{y_m^{H}(t), y_{m-1}^{SN}(t)\}$.
        $c_m \leftarrow \tau$.
      end if
    end for
    $t \leftarrow t + 1$.
  end for
  return $O = \frac{y_L^{SN}(t)}{O_{\max}}$.
End Algorithm
```

C. LTM Storage Mechanism

This section presents the storage mechanism of LTM cells. For a given training sequence $\mathbf{S}$, existing LTM cells compete for the best match to determine the winning sequence by a winner-take-all (WTA) network. If the match is sufficient, i.e., the matching signal is smaller than a predefined threshold θ, the corresponding winning LTM cell plays its sequence and no learning occurs. However, if the match is not sufficient, a learning signal is triggered and a new LTM cell is employed to learn the input sequence as corresponding synaptic weight $\mathbf{W}$ using one-shot learning.

In an intelligent system, the threshold θ of an LTM cell is determined via its interaction with the environment and is task-dependent. In this paper, we set the learning threshold theta to zero (in the training phase). Therefore, an LTM cell is dedicated separately to each input sequence. One-shot learning is used in many neural systems [21], [33], [34] to improve the training efficiency of the learning system. The key idea is to set the learning rate to be very high to imprint the sequence to an LTM cell by a single observation of the sequence. The learning of a pattern proceeds in a sequential fashion with a sequence counter P [3]. At each activation of the jth sequence counter, the jth element of the sequence is mapped to the connection between the input layer and the jth PN. Distributed representation of LTM cells can be incorporated to improve the storage capacity. But in this paper, for reasons of simplicity, we only considered a localist representation of sequences.

D. Sequence Recognition Algorithm

This section develops a sequence recognition algorithm called LTMSR (Algorithm 1) based on the architecture shown in the Fig. 1. Each input vector of a test sequence is incrementally presented to an LTM cell. Once the matching output is returned, a winning LTM sequence can be determined by a WTA network of the existing LTM cells. The LTMSR introduces the delay factor τ and corresponding counters $\mathbf{C} \equiv \{c_j | j = 1, \ldots, L\}$, which retain the SNs' activations

TABLE I

OUTPUT OF THE LTM CELL WITH VARIOUS INPUTS. T: PERTURBATION TYPE, NO: NORMALIZED OUTPUT, UO: UNNORMALIZED OUTPUT

Input	T	NO	UO	Input	T	NO	UO
ABCD	0	1.00	3.20	BCD	3	0.75	2.40
ABDC	1	0.75	2.40	BC	3	0.50	1.60
ACBD	1	0.69	2.20	AB	3	0.50	1.60
CBAD	1	0.50	1.60	A	3	0.25	0.80
ADCB	1	0.38	1.20	B	3	0.25	0.80
DCBA	1	0.25	0.80	WN 1	4	0.99 (±0.00)	3.18
ABBCD	2	0.94	3.00	WN 2	4	0.86 (±0.06)	2.74
ABCCD	2	0.94	3.00	WN 3	4	0.57 (±0.15)	1.83
ABBBCD	2	0.88	2.80	WN 4	4	0.39 (±0.17)	1.24
ABCCCD	2	0.88	2.80	WN 5	4	0.29 (±0.16)	0.93
ACD	3	0.75	2.40	WN 6	4	0.13 (±0.12)	0.42

for a number of steps before being reset. The purpose of the delay factor is to compensate for minor delay or perturbation of input **I**. The computational complexity of the algorithm is in the order of O(L), where L is the length of the LTM cell for each input vector. An example of the network operation given a sequence is described next.

Example 1: We consider the 2-D sequence $\mathbf{S} \equiv ABCD$ with the length of 4. Each element of the sequence is specified as follows: $A = (0.2, 0.8), B = (0.4, 0.6), C = (0.6, 0.4)$, and $D = (0.8, 0.2)$. The sequence **S** is stored as an LTM cell by one-shot learning. Therefore, there are two neurons in the input layer and four neurons in each of the primary, intermediate, and secondary layers. The specifications of the LTM cell are set as follows: $\delta_{ij} = 0.1, \forall i \in [1, 2], j \in [1, 4], \phi_j = 1.0, \forall j \in [1, 4]$, and $\tau = 1$.

A number of test sequences are synthesized based on the stored sequence to evaluate the robustness of the LTM cell's activation. The result is shown in Table I. The original sequence (Type 0) and four types of sequential distortions, including order distortion (Type 1), replicated elements (Type 2), missing elements (Type 3), and noisy elements (Type 4) are introduced. The noisy test sequences are generated by adding white noise (with zero mean and standard deviation σ) to the original sequence. The values of σ are 0.01, 0.05, 0.1, 0.15, 0.2, and 0.3, which correspond to the test sequences WN 1–WN 6 in Table I. The simulations with noisy sequences were conducted with 1000 random trials for each σ. The average outputs with unnormalized (absolute) activations and normalized (absolute values divided by $O_{\max}$) activations are reported. The decay parameter γ is set to 0.2.

The first observation is that the original sequence elicits the maximum activation ($O_{\max} = 3.2$) among all the cases. This verifies the correctness of the Lemma 1. Second, for each type of distortion, the output of the LTM cell reflects an increase of the distortion level by graceful degradation of activation.

E. Error-Tolerance

This section proposes an adaptive characterization of uncertainties based on the local variation of features. The estimated uncertainty is used as the tolerance Δ in (2) to normalize the matching between LTM elements and an input vector. The proposed mechanism is appropriate for spatio-temporal patterns where local variations in time provide useful information for uncertainty analysis.

Given a synaptic connection **W** of an LTM cell, the task of uncertainty estimation is to characterize the local standard deviation (LSD) of elements with respect to the temporal axis. The LSD is estimated over a local window $\Psi_m (m \in [1, L])$ of size $2\tau_\Psi + 1$ (τ_Ψ is an integer) centered at the mth element of the sequence. The two boundaries of the sequence are mirror-extended by τ_Ψ components to allow a full-sized envelope when $m \in [1, \tau_\Psi]$ or $[L - \tau_\Psi, L]$. If we denote $B = (2\tau_\Psi + 1)^{-1}$, then the LSD of the ith feature of the mth element is given by

$$\delta_{im} = \sqrt{B \sum_{j \in \Psi_m} (w_{ij} - \mu_{im})^2} \tag{19}$$

where μ_{im} is the mean of the ith feature with respect to the local window

$$\mu_{im} = B \sum_{j \in \Psi_m} w_{ij}. \tag{20}$$

The estimation of **D** can be performed by a sliding window over the temporal axis. Therefore, a moving-window approach to LSD estimation can be applied.

Corollary 1: The incremental update of the LSD for a sequence **S** is

$$\mu_{im} = \begin{cases} B \sum_{j \in \Psi_1} w_{ij}, & \text{if } m = 1 \\ \mu_{im-1} + B(x_r - x_l), & \text{otherwise} \end{cases} \tag{21}$$

$$\delta_{im} = \begin{cases} \sqrt{B \sum_{j \in \Psi_j} (w_{ij} - \mu_{im})^2}, & \text{if } m = 1 \\ \sqrt{\delta_{im-1}^2 + B(x_r^2 - x_l^2) - (\mu_{im}^2 - \mu_{im-1}^2)}, & \text{otherwise} \end{cases} \tag{22}$$

where $x_l = w_{im-\tau_\Psi-1}$ and $x_r = w_{im+\tau_\Psi}$.

Proof:

1) $m = 1$: Equations (21) and (22) simply follow the original formulae (19) and (20), respectively.

2) $m > 1$: From (20), we have

$$\begin{aligned} \mu_{im} &= B\left(\sum_{j \in \Psi_{m-1}} w_{ij} + x_r - x_l\right) \\ &= \mu_{im-1} + B(x_r - x_l). \end{aligned} \tag{23}$$

Thus follows (21). From (19), for all m we have

$$\begin{aligned} \delta_{im}^2 &= B \sum_{j \in \Psi_m} (w_{ij} - \mu_{im})^2 \\ &= B \sum_{j \in \Psi_m} w_{ij}^2 - 2B\mu_{im} \sum_{j \in \Psi_m} w_{ij} + \mu_{im}^2 \\ &= B \sum_{j \in \Psi_m} w_{ij}^2 - 2\mu_{im}^2 + \mu_{im}^2 \\ &= B \sum_{j \in \Psi_m} w_{ij}^2 - \mu_{im}^2. \end{aligned} \tag{24}$$

Similarly, we have

$$\delta_{im-1}^2 = B \sum_{j \in \Psi_{m-1}} w_{ij}^2 - \mu_{im-1}^2. \tag{25}$$

Subtract (25) from (24) and note that $\sum_{j\in\Psi_m} w_{ij}^2 - \sum_{j\in\Psi_{m-1}} w_{ij}^2 = x_r^2 - x_l^2$ we have

$$\delta_{im}^2 = \delta_{im-1}^2 + B(x_r^2 - x_l^2) - (\mu_{im}^2 - \mu_{im-1}^2).$$

After taking the square roots of both sides, we have (22). ■

Corollary 1 suggests that when $m > 1$, the LSD of the mth element can be updated from that of the $(m-1)$th element based on the right-most and left-most element of the moving window. Thus, it is more computationally efficient than the direct estimation method in (19) that involves all the elements of the window at each time. In this paper, the tolerance estimation is performed with each feature individually. The influence of covariance of features toward tolerance estimation is under investigation. In the following experiments, τ_Ψ is set to five unless otherwise stated.

F. Significance

Significance analysis provides an evaluation of the importance of each element within an LTM cell, which helps the LTM cell to focus on identifying highly distinguishing elements in a sequence. In this paper, the significance of elements Φ in the sequence is integrated to modulate the activation of an LTM cell. Previous works in [3] and [21] give equal emphasis to all the elements within the sequence. An important objective of this paper is to leverage the role of significance, and to verify its impact on improving the recognition performance of a sequence recognizer.

The significance estimation of elements proceeds from the feature level to the element level. The significance estimation is based on statistical characteristics of the features' values throughout the temporal domain. Given an LTM cell, we denote the mean and standard deviation of the ith feature $(i = 1, \ldots, F)$ as μ_i and σ_i, respectively. They are empirically computed as follows:

$$\mu_i = \frac{1}{L}\sum_{j=1}^{L} w_{ij} \tag{26}$$

$$\sigma_i = \sqrt{\frac{1}{L-1}\sum_{j=1}^{L}(w_{ij} - \mu_i)^2}. \tag{27}$$

The significance estimation of an LTM cell at the feature level is denoted as $\mathbf{R} \equiv \{r_{ij} | i = 1, \ldots, F, j = 1, \ldots, L\}$ and is computed as

$$r_{im} = 1 - \exp\left\{-\frac{(w_{im} - \mu_i)^2}{\eta\sigma_i^2}\right\}, \quad m = 1, \ldots, L \tag{28}$$

where η is a tuning parameter. Subsequently, the significance estimation of the LTM cell at the element level, i.e., Φ, is computed as

$$\phi_m = \sqrt{\frac{\sum_{k=1}^{F} r_{km}^2}{F}}, \quad m = 1, \ldots, L. \tag{29}$$

From (28), we have $r_{ij} \in [0, 1], \forall i, j$, therefore, $\phi_j \in [0, 1], \forall j$.

Intuitively, the significance estimation based on (29) gives high significance values to the elements in which the feature values are statistically different from the mean values, and low significance to the elements in which the feature values are close to the mean values. It must be highlighted that the proposed significance estimation method was found suitable for our chosen applications but may need to be re-formulated for other domains with different data characteristics.

III. Statistical Analysis of LTM Cell Properties

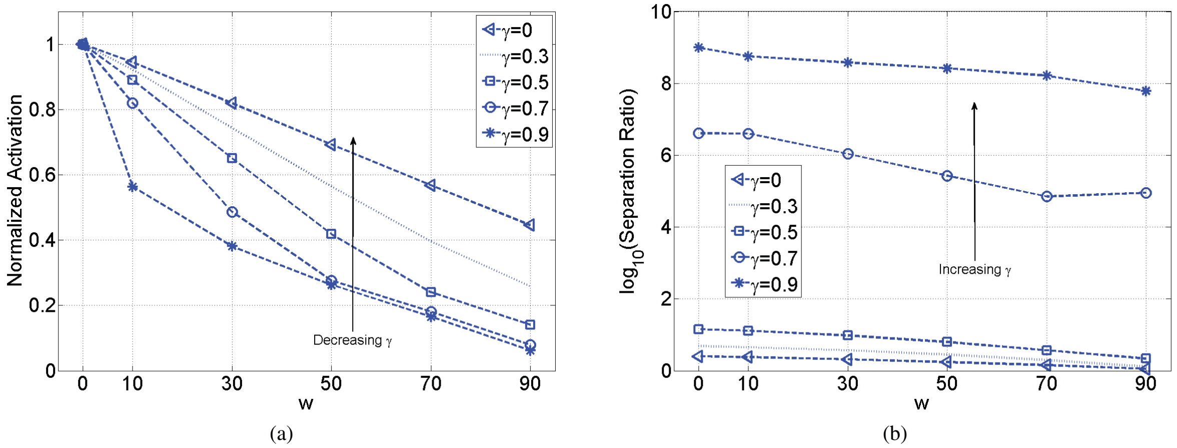
In this section, various properties of an LTM cell are evaluated with a series of synthetic simulations. Random multidimensional sequences were generated and learnt as LTM cells. Statistical characteristics of the proposed LTM model were empirically investigated with synthesized test sequences, which contain various types and magnitudes of sequential distortions from the original sequences. The evaluation methodology is based on three criteria. The first criterion is the prediction accuracy (PA) that is defined by the number of correct predictions of test sequences (indicated by the strongest response from the LTM cells) divided by the total number of test sequences. The second criterion is the strength of the normalized activation (NA) provided by the winning LTM cell. The third criterion is the separation ratio (SR) that is defined by the ratio between the best and the second-best LTM output when a correct prediction is obtained. The motivation of SR estimation is to quantify the strength of the decision made by the winning LTM.

We first evaluate the error tolerance of the LTM model with four different types of error as in Example 1. Subsequently, the influence of significance of sequence elements was investigated. For each type of experiments, ten different simulations were conducted and the average results were reported. In each simulation, four 10-D sequences, denoted as $P^i, (i = 1, \ldots, 4)$, with the length of 100 were generated from uniform distribution in [0, 1]. Each P^i was then stored as a separate LTM cell and various test sequences were synthesized from P^i to evaluate the performance.

A. Tolerance With Error Type 1 - Local Order Perturbation

In order to evaluate the impact of local order perturbation, the test sequences were produced as follows. For each P^i, a local window that occupies $w(\%)(0 \le w \le 100)$ of P^i was randomly extracted, permuted, and then placed back to P^i to construct a test sequence. For each selected w and P^i, 1000 test sequences were generated. During testing, a test sequence was classified as P^i if the corresponding LTM cell elicits the maximum NA among all LTM cells. The influence of the decay rate was also taken into account as different values of $\gamma = 0.0, 0.1, 0.3, \ldots, 0.9$, were considered. Other LTM parameters were set to $\tau = 0$ and $\eta = 0$. The setting of η effectively makes $\phi_m = 1, \forall m$ in this experiment.

Results in Table II show that the LTM model maintains a consistent 100% recognition accuracy for all chosen γs. Fig. 3(a) shows that the NA of the LTM cells decreases



(a)

(b)

Fig. 3. (a) NA and (b) $\log_{10}$(SR) of LTM activations with different amount of local order perturbations in the case of error type 1.

TABLE II

MEAN AND STANDARD DEVIATION OF PA IN THE CASE OF ERROR TYPE 1. NOTE THAT $w = 0$ WHEN THERE WAS NO DISTORTION

w	0	10	30	50	70	90
PA	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.00)	1.0 (±0.0)

TABLE III

MEAN AND STANDARD DEVIATIONS PA OF THE LTM CELLS WITH DIFFERENT DELAY FACTORS IN THE CASE OF ERROR TYPE 2

q	$\tau = 0$	$\tau = 5$	$\tau = 10$	$\tau = 15$	$\tau = 20$	$\tau = 25$
5	0.70 (±0.06)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)
10	0.71 (±0.10)	1.00 (±0.05)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)
15	0.75 (±0.08)	0.75 (±0.04)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)
20	0.63 (±0.14)	0.75 (±0.04)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)
25	0.62 (±0.19)	0.80 (±0.04)	0.88 (±0.10)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)

gradually as the amount of perturbation increases. Second, the activation of the LTM cell with a similar amount of perturbation decreases as γ increases. By increasing the decay rate, we also obtained a better separation margin as in Fig. 3(b) without recognition performance degradation. In real applications, the value of γ should be optimized for desirable performance by a cross validation technique for example.

B. Tolerance With Error Type 2 - Replicated Elements

In order to examine the performance of the LTM model when part of a learnt sequence is replicated, the test sequences were generated as follows. For each P^i, 50 elements at random locations were selected. At each location, the respective element was replicated q times to form a test sequence with length longer than the original training sequence. Hundred test sequences were constructed for each P^i and selected q with different replicated locations. The value of q was varied from 5 to 20 with an increasing step of 5. Other LTM parameters were set to $\gamma = 0$ and $\eta = 0$.

Table III shows the average PA with ten different runs of simulations. In this case, the delay factor τ was varied from 0 to 25 with an increasing step of 5. By increasing the delay factor τ, the PA was improved and 100% PA was achieved as soon as τ reached q. Fig. 4(a) and (b) shows the NA and SR values with different amount of delay distortions. By increasing τ, both NA and SR were enhanced. Additionally, the activation of LTM attained its maximum level, i.e., 1.0, when the delay factor coincided with the number of replicated elements. Similarly, the SR achieved its maximum value when τ is equal to q.

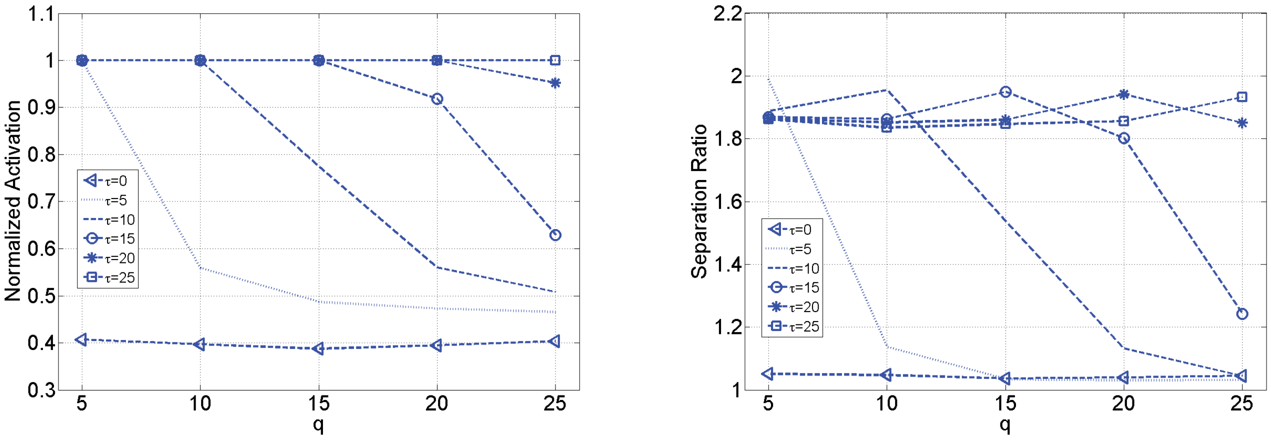
C. Tolerance With Error Type 3 - Missing Elements

In order to examine the performance of the LTM model when part of a learnt sequence is missing, the test sequences were generated as follows. For each P^i, $p(\%)$ $(0 \leq p \leq 100)$ of the elements at different locations were removed to form a test sequence. Five different values of $p = 20, 40, 60, 80, 90$, were selected and 1000 test sequences were generated for each p and P^i. Various LTM parameters were set to $\gamma = 0, \tau = 0$, and $\eta = 0$.

Table IV shows the average PA with ten different simulations. We achieved a perfect recognition rate for every selected p. Fig. 5 shows that by increasing p, both NA and SR were reduced. Due to the linear decay function and equal significance among elements, a similar number of missing elements at arbitrary locations result in a similar reduction of LTM activation. Therefore, a linear reduction curve of LTM activation was obtained with a zero standard deviation.

D. Tolerance With Error Type 4 - Noisy Elements

This section examines the performance of the LTM model when the elements of the sequence were noisy. During training, each sequence P^i was added with WGN with standard deviation of 0.1 and then learnt by an LTM cell. The test sequences were generated by adding WGN with variable standard deviations κ to the training sequences. Six different



(a) (b)

Fig. 4. (a) NA and (b) SR of LTM activations with different number of replicated elements in the case of error type 2.

TABLE IV
PA of the LTM Cells With Different Amounts of Missing Elements in the Case of Error Type 3

p	20	40	60	80	90
PA	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)

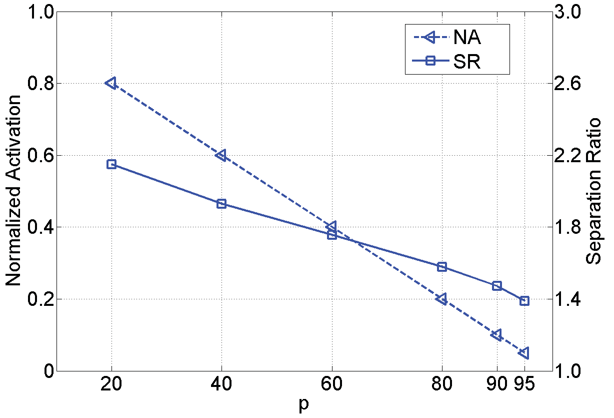


Fig. 5. NA and SR of LTM activations with different numbers of missing elements in the case of error type 3.

values of $\kappa = 0.1, 0.2, 0.4, \ldots, 1.0$ were selected and 1000 test sequences were generated for each κ and P^i. Various LTM parameters were set to $\gamma = 0, \tau = 0$, and $\eta = 0$.

Table V shows the PA of ten different runs of simulations. Three different window sizes (2, 5 and 100) of the procedure described in Section II.E were selected. We achieved perfect recognition rates for all selected window sizes until κ is 0.6 which is 6 times the noise in the LTM cells. Second, by having $\tau_\Psi = 5$, the performance was improved compared to a small window size $\tau_\Psi = 2$ and not significantly different from the largest possible window size $\tau_\Psi = 100$, which is the length of the training sequence. Fig. 6(a) and (b) shows that both NA and SR reduced gracefully with increasing κ.

TABLE V
PA of the LTM Cells With Different Standard Deviations of WGN in the Case of Error Type 4 ($\gamma = 0$)

τ_Ψ	$\kappa = 0.1$	$\kappa = 0.2$	$\kappa = 0.4$	$\kappa = 0.6$	$\kappa = 0.8$	$\kappa = 1.0$
2	1.000 (±0.00)	1.000 (±0.00)	1.000 (±0.00)	0.980 (±0.01)	0.525 (±0.02)	0.310 (±0.01)
5	1.000 (±0.00)	1.000 (±0.00)	1.000 (±0.00)	1.000 (±0.00)	0.768 (±0.01)	0.395 (±0.00)
100	1.000 (±0.00)	1.000 (±0.00)	1.000 (±0.00)	1.000 (±0.00)	0.783 (±0.02)	0.402 (±0.01)

E. Influence of Significance of Elements

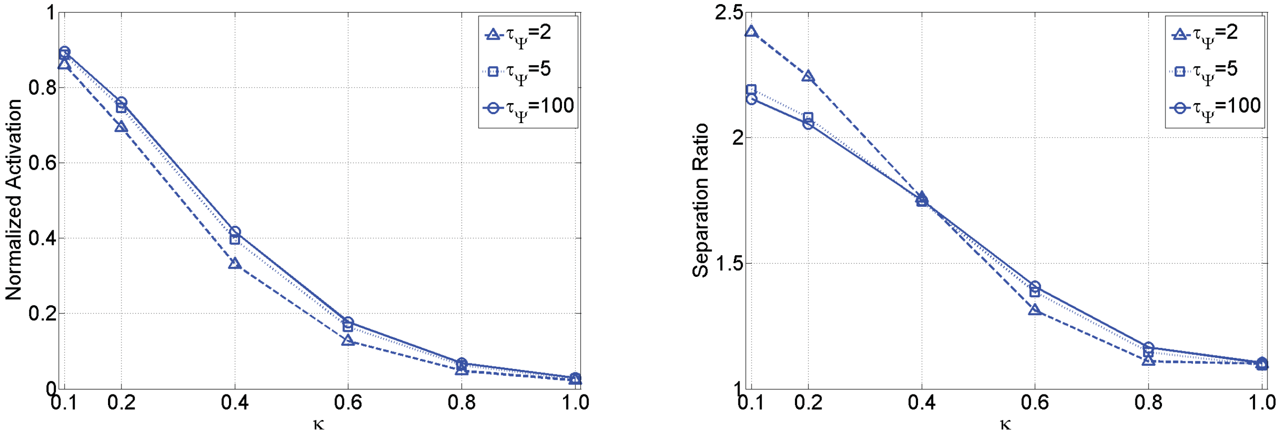
In order to exploit the role of significance of elements, the following 2-D sequential pattern, denoted as $\mathbf{Q} = \{q_{ij} | i = 1, 2;\ j = 1, \ldots, 100\}$, was considered:

$$\mathbf{Q} : q_{1j} = \begin{cases} \frac{(j-40)}{20}, & \text{if} \ \ 41 \leq j \leq 60 \\ 0, & \text{otherwise} \end{cases} \qquad (30)$$

$$q_{2j} = \begin{cases} 1, & \text{if} \ \ 41 \leq j \leq 60 \\ 0, & \text{otherwise.} \end{cases} \qquad (31)$$

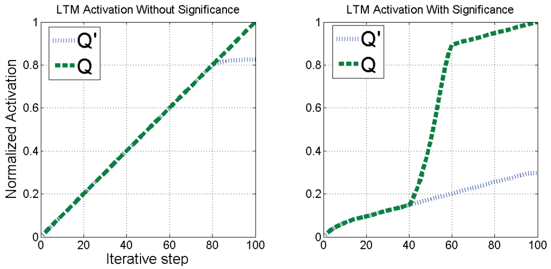
The pattern $\mathbf{Q}$ is assumed to contain 80% insignificant feature (zero values) and 20% significant feature (nonzero values) in each of the dimension. The significance estimation described in Section II.F gives high values to the salient elements and low values to the non-salient elements. It is expected that by incorporating the significance of elements, the LTM activation shows a clear separation between the pattern $\mathbf{Q}$ and a pattern that contains purely insignificant features.

Fig. 7 illustrates a case when $\mathbf{Q}$ was stored as an LTM cell and the two test sequences were $\mathbf{Q}$ itself, and a sequence $\mathbf{Q}'$ of a similar length that contains only zero-value features. By the end of the test sequences, a successful recognition of $\mathbf{Q}$ was evident due to the stronger response of the sequence $\mathbf{Q}$ compared to $\mathbf{Q}'$. More importantly, the separation margin between the two test sequences was clearly improved when the significance of sequence elements is incorporated [Fig. 7(b)]. This is in contrast to the case when every element of the LTM cell is allocated a similar significance value [Fig. 7(a)].



(a) (b)

Fig. 6. (a) NA and (b) SR of LTM activations with different standard deviations of WGN in the case of error type 4.



(a) (b)

Fig. 7. NA of an LTM cell that stores the sequence Q with two test signals Q and Q′ (a) without and (b) with the influence of significance ($\eta = 4.0$) of elements.

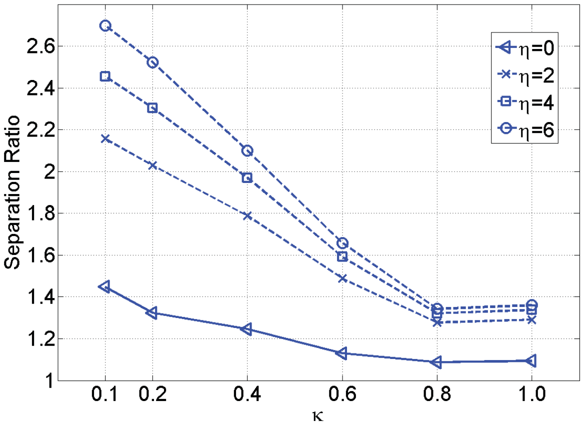


Fig. 8. Influence of the significance of elements to the SR of T1 and T2 with different noise levels.

To verify the hypothesis, the simulation was designed as follows. First, the sequence **Q** corrupted with additive WGN of the standard deviation 0.1 was stored as an LTM sequence. Various parameters of the LTM cell were set to $\gamma = 0$ and $\tau = 0$. Two types of test sequences were generated. The first type (T1) was constructed by adding WGN of variable standard deviation κ to the sequence **Q**. The values of κ were similar to those used in the Section II.D. The second type (T2) was a sequence that is of the same length as **Q** but contains *only* WGN with variable standard deviation κ. For each chosen κ, 1000 runs were conducted. In each run, a sequence of type T1 and T2 was generated and their respective SRs between the LTM activations were computed. Fig. 8 shows the average SR of the 1000 runs for each chosen κ. The results show that by incorporating the significance of elements into the modulation of LTM activation (i.e., $\eta > 0$), the SR was consistently improved from the other case (i.e., $\eta = 0$). This observation confirms the advantage of incorporating significance of elements in discriminating sequential patterns provided an estimation of Φ.

IV. Experiments With ASL Dataset

The ASL dataset [32] contains samples recorded by a high-quality hand position tracker from a native signer expressing various Auslan signs. The total number of signs are 95, each of which was recorded 27 times, organized in nine different sessions. Each sample contains a 27-D temporal pattern of average length 57. In this paper, we investigate the capability of the proposed LTM model in classifying trajectory given a hand sign sample. We use the first-order derivatives of the x and y coordinates of both hands (four dimensions) as the feature set. Additionally, each dimension of the extracted trajectories is low-pass filtered by a moving average window of size three.

A. Experimental Design and Analysis

For the purpose of proper comparison, similar experimental setup as in [35] was used. We used half of the trajectories (i.e., 13 samples per sign) as the training set, and all the available trajectories as the testing set (i.e., 27 samples per sign). During training, each of the samples of the training set was stored as a separate LTM cell with the label of the corresponding hand sign. To achieve a desirable performance, three parameters needed to be optimized, namely the decay rate γ, significance factor η, and the delay factor τ. The window size τ_Ψ for LSD estimation was set to ten. Empirical studies suggest that a bigger window did not produce significantly different results. For decision making, a test sample was assigned to the sign

TABLE VI
VALIDATION RESULTS (PA, NA, AND SR) OF THE LTM MODEL DURING THE TRAINING STAGE FOR THE ASL DATASET

No. of classes (no. of runs/no. of training instances)	**PA**	**NA**	**SR**
8 (50/104)	0.8772 (± 0.0492)	0.5462 (± 0.1334)	1.4153 (± 0.4211)
16 (30/208)	0.8268 (± 0.0332)	0.5828 (± 0.1073)	1.4601 (± 0.4036)
29 (10/377)	0.7942 (± 0.0316)	0.5819 (± 0.1054)	1.4180 (± 0.3721)
38 (10/494)	0.7799 (± 0.0171)	0.5554 (± 0.1335)	1.3878 (± 0.3486)

of the maximum activated LTM cells, and a correct prediction was counted if the assigned label coincides with the true label of the sample.

The parameters were optimized by a standard three-fold cross validation on the training set in the grid specified by $\gamma \in [0.0, 0.1, \ldots, 1.0]$, $\eta \in [0, 1, 2, \ldots, 16]$ and $\tau \in [0, 1, 2, \ldots, 20]$. We performed experiments with four different number of selected classes i.e., 8, 16, 29, and 38. For a number of classes C, we repeatedly collected samples from C random signs of the total 95 signs for multiple runs.

To quantify the results, we used three criteria as in Section III: PA of classification, NA of the winning LTM cells, and SR. Given a correct classification of a sample, the SR in this case was computed slightly differently as the ratio between the activation of the winning LTM cell and the highest activated LTM cell that belongs to a different class. Table VI shows the result during the validation stage. It can be seen that the accuracy gradually reduces as the number of classes increases. Second, the NA and SR of the winning LTM cells are relatively stable across multiple runs and numbers of classes.

To elucidate the sensitivity of the proposed LTM model to the different parameters, for a selected number of classes C, we first obtained the optimal parameters $(\gamma^*, \eta^*, \tau^*)$ by cross validation. Subsequently, two of the optimal parameters were fixed while the third one was varied. The average results for $C = 16$ in 30 different runs were plotted in Fig. 9.

The first observation is that the performance in terms of PA was consistently improved when each of the parameters was incorporated (by setting each parameter to be positive). The improvement of PA with the modulation of significance (i.e., $\eta > 0$) reveals that the proposed significance estimation is appropriate in assisting sign language's interpretation. The second observation is that an improvement of PA was obtained when SR was improved. The only exception is when the decay rate γ is very high ($\approx$1.0). In this case, a perfect recognition of an element of the sequence typically leads to only a little gain of activation. This results in a weak LTM activation that translates into high decision making ambiguity. We also found empirically that the performance of the model saturated in terms of PA when $\tau \geq 6$ or $\eta \geq 4$. Similar observations were obtained for different numbers of selected classes.

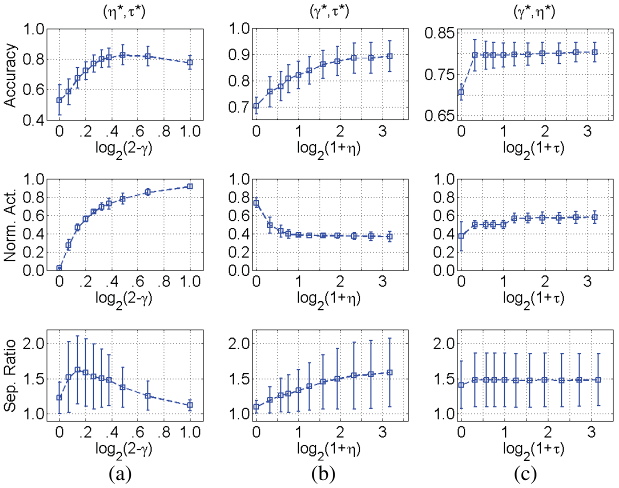


Fig. 9. Sensitivity of the LTM cells to the varying parameters in a 16-class problem. Sensitivity of PA, NA, and SR (from top to bottom) to the varying (a) γ, (b) η, and (c) τ. Note that the horizontal axes are plotted in log-scale.

TABLE VII
PA OF THE LTM MODEL AND COMPARISON WITH OTHER LEARNING MODELS FOR THE ASL DATASET

Model		**No. of classes (no. of runs/no. of test instances)**			
		8 (50/216)	**16 (30/432)**	**29 (10/738)**	**38 (10/1026)**
Proposed LTM model	$(\gamma = 0, \eta^*, \tau^*)$	0.9257 (± 0.0368)	0.8948 (± 0.0324)	0.8506 (± 0.0305)	0.8305 (± 0.0225)
	$(\gamma^*, \eta = 0, \tau^*)$	0.7960 (± 0.0434)	0.7480 (± 0.0346)	0.7056 (± 0.0232)	0.6899 (± 0.0124)
	$(\gamma^*, \eta^*, \boldsymbol{\tau = 0})$	0.8719 (± 0.0360	0.8354 (± 0.0236)	0.7874 (± 0.0203)	0.7704 (± 0.0186)
	$(\gamma^*, \eta^*, \tau^*)$	**0.9412** (± 0.0244)	**0.9009** (± 0.0296)	**0.8884** (± 0.0189)	**0.8671** (± 0.0147)
Original LTM model [3]		0.8102 (± 0.0409)	0.7676 (± 0.0200)	0.7372 (± 0.0135)	0.7263 (± 0.0346)
HMM [35]		0.86	0.78	0.69	0.66
GMM [35]		0.85	0.74	0.67	0.64
SOM [36]		0.82	0.76	NA	NA

B. Results and Comparisons With Other Models

In this section, we benchmark the performance of the LTM model with other published works. Classification accuracy is reported following the protocol in [35]. The performance of the LTM model was compared with the original LTM model [3], HMM [35], Gaussian mixtures model (GMM) [35], and self-organizing map (SOM) [36] for a similar task. The model in [3] that was developed for character sequence processing can be obtained from the model in this paper by "turning off" the influence of delay factor and significance of elements, i.e., setting η and τ to 0. The decay rate γ for the model was optimized by cross validation as described in Section IV.B.

TABLE VIII
PA OF THE LTM MODEL WITH VARIABLE SIZE OF TRAINING SET

Number of classes	**Percentage of the original training set (%)**				
	20	**40**	**60**	**80**	**100**
8	0.6708 (± 0.0668)	0.8106 (± 0.0595)	0.8528 (± 0.1085)	0.9269 (± 0.0587)	0.9412 (± 0.0244)
16	0.5255 (± 0.0619)	0.7150 (± 0.0829)	0.8310 (± 0.0305)	0.8981 (± 0.0307)	0.9009 (± 0.0296)
29	0.4918 (± 0.0506)	0.6653 (± 0.0467)	0.8059 (± 0.0279)	0.8163 (± 0.0180)	0.8884 (± 0.0189)
38	0.4284 (± 0.0530)	0.6239 (± 0.0484)	0.7585 (± 0.0198)	0.7928 (± 0.0196)	0.8671 (± 0.0147)

The result is tabulated in Table VII. The effect of parameters of the proposed LTM model was experimented by setting one parameter to 0 individually while fixing the other two at their optimal values. It can be observed that the proposed LTM model significantly outperformed other learning models in all selected number of classes. Second, the roles of the introduced parameters were clearly substantiated by an improvement of PA when each of them was employed.

Another interesting investigation is to determine the sensitivity of the LTM model to the number of training instances. In this experiment, the fraction of the original training set used to construct LTM cells was varied. The result on a similar testing set is reported in Table VIII. We report two important observations: 1) the performance was improved when the number of training instances increased and 2) the LTM model surpassed the performance of [35] with fewer number of training examples for all selected classes. For instance, the performance with 38 classes in [35] can be obtained with only 60% of the training set with the proposed LTM model. These observations highlight the advantage of robust and reliable storage properties of the proposed LTM structure.

V. CONCLUSION

In this paper, we described a neural network approach to temporal sequence learning, memory organization, and recognition. The main characteristics of the model include the LTM organization of multidimensional real-valued sequence, the robust matching algorithm with error tolerance, the significance of sequence elements, and the memory forgetting mechanism. Our analytical approach to the LTM activation's normalization allows the comparison of sequences of different lengths. The merits of the proposed framework were demonstrated using a series of synthetic simulations and the ASL dataset. It was believed that the proposed architecture is general enough to support different types of applications that require complex sequential processing in perception, prediction, and motor control. Additional applications in cognitive activities and speech recognition are currently being explored.

A vital future of the proposed LTM model is an efficient sequence alignment scheme for combining multiple sequences of a similar content. This characteristic can provide a compact representation when a large number of training sequences are present. It is also useful to learn spatio-temporal structures from many sequences to enhance the error-tolerance capability.

One of the main uses of the proposed LTM model is to construct a stable and reliable self-organizing structure of EM [11]. EM allows an embodied intelligent agent to remember and re-experience previously acquired sequences of events, i.e., episodes. Therefore, this paper has presented an essential building block in the development of cognitive machines [37], [38].

ACKNOWLEDGMENT

The authors would like to thank the three anonymous reviewers for providing valuable comments to this paper.

REFERENCES

[1] R. O'Reilly and Y. Munakata, *Computational Explorations in Cognitive Neuroscience*. Cambridge, MA: MIT Press, 2000.

[2] 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.

[3] J. A. Starzyk and H. He, "Spatio–temporal memories for machine learning: A long-term memory organization," *IEEE Trans. Neural Netw.*, vol. 20, no. 5, pp. 768–780, May 2009.

[4] 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.

[5] R. Sun and C. L. Giles, *Sequence Learning: Paradigms, Algorithms and Applications*. New York: Springer-Verlag, 2001.

[6] S. C. Kremer, "Spatiotemporal connectionist networks: A taxonomy and review," *Neural Comput.*, vol. 13, no. 2, pp. 249–306, Feb. 2001.

[7] J. McGaugh, "Memory–a century of consolidation," *Science*, vol. 287, no. 5451, pp. 248–251, Jan. 2000.

[8] J. J. Todd and R. Marois, "Capacity limit of visual short-term memory in human posterior parietal cortex," *Nature*, vol. 428, no. 6984, pp. 751–754, Apr. 2004.

[9] D. L. 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.

[10] D. O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

[11] L. Shastri, "Episodic memory trace formation in the hippocampal system: A model of cortico-hippocampal interaction," Int. Computer Science Inst., Berkeley, CA, Tech. Rep. TR-01-004, 2001.

[12] S. Grossberg, "Some networks that can learn, remember and reproduce any number of complicated space-time patterns," *J. Math. Mech.*, vol. 19, no. 1, pp. 53–91, Jul. 1969.

[13] G. Bradski, G. A. Carpenters, and S. Grossberg, "Store working memory networks for storage and recall of arbitrary temporal sequences," *Biol. Cybern.*, vol. 71, no. 6, pp. 469–480, 1994.

[14] K. J. Lang, A. H. Waibel, and G. E. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Netw.*, vol. 3, no. 1, pp. 23–43, 1990.

[15] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Apr.–Jun. 1990.

[16] M. I. Jordan, "Serial order: A parallel distributed processing approach," Inst. Cognitive Science, Univ. California, Berkeley, Tech. Rep. AD-A-173989/5/XAB, 1986.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[20] J. Martens and I. Sutskever, "Learning recurrent neural network with Hessian-free optimization," in *Proc. Int. Conf. Mach. Learn.*, Bellevue, WA, 2011, pp. 1–8.

[21] D. L. 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.

[22] D. L. Wang and B. Yowono, "Anticipation-based temporal pattern generation," *IEEE Trans. Syst., Man Cybern.*, vol. 25, no. 4, pp. 615–628, Apr. 1995.

[23] L. Wang, "Learning and retrieving spatio-temporal sequences with any static associative neural network," *IEEE Trans. Circuits Syst. II, Analog Digital Signal Process.*, vol. 45, no. 6, pp. 729–739, Jun. 1998.

[24] L. Wang, "Multi-associative neural networks and their applications to learning and retrieving complex spatio-temporal sequences," *IEEE Trans. Syst., Man Cybern., Part B, Cybern.*, vol. 29, no. 1, pp. 73–82, Feb. 1999.

[25] J. Hawkins and D. George. *Hierarchical Temporal Memory: Concepts, Theory and Terminology*. Numenta, Inc., Menlo Park, CA [Online]. Available: http://www.numenta.com/

[26] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[27] A. Pronobis, B. Caputo, P. Jensfelt, and H. Christensen, "A realistic benchmark for visual indoor place recognition," *Robot. Auton. Syst.*, vol. 58, no. 1, pp. 81–96, Jan. 2010.

[28] D. Liu, X. Xiong, B. DasGupta, and H. Zhang, "Motif discoveries in unaligned molecular sequences using self-organizing neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 919–928, Jul. 2006.

[29] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 171–177, Jan. 2010.

[30] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Vis. Res.*, vol. 49, no. 10, pp. 1295–1306, Jun. 2009.

[31] A. Arleo, F. Smeraldi, and W. Gerstner, "Cognitive navigation based on nonuniform Gabor space sampling, unsupervised growing networks, and reinforcement learning," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 639–652, May 2004.

[32] M. W. Kadous, "Temporal classification: Extending the classification paradigm to multivariate analysis," Ph.D. thesis, School Comput. Sci. Eng., Univ. New South Wales, Kensington, Australia, 2002.

[33] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vis. Image Process.*, vol. 37, no. 1, pp. 54–115, Jan. 1987.

[34] A. T. L. Phuan, J. M. Zurada, W. L. Ping, and J. Xu, "The hierarchical fast learning artificial neural network (HieFLANN)–an autonomous platform for hierarchical neural network construction," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1645–1657, Nov. 2007.

[35] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden Markov models," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1912–1919, Jul. 2007.

[36] A. Naftel and S. Khalid, "Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space," *Multimedia Syst.*, vol. 12, no. 3, pp. 227–238, 2006.

[37] J. A. Starzyk, "Motivation in embodied intelligence," in *Frontiers in Robotics, Automation and Control*. Vienna, Austria: I-Tech Publishing, 2008, pp. 83–110.

[38] J. Bach, *Principles of Synthetic Intelligence: An Architecture for Motivated Cognition*. New York: Oxford Univ. Press, 2009.

Vu Anh Nguyen (S'09) was born in Hanoi, Vietnam, in 1987. He received the B.E. degree (with honors) from the School of Computer Engineering, Nanyang Technological University, Singapore, where he is currently pursuing the Ph.D. degree.

He has been excited about investigating cognitive autonomous systems specialized in spatiotemporal memory structures and hierarchical organization. His research is always augmented with real-world implementation and applications. His current research interests include cognitive visual understanding, cognitive neural networks, computer vision, robotic navigation, and motivated learning.

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

He was an Assistant Professor with the Institute of Electronics Fundamentals, Warsaw University of Technology. Subsequently, he spent two years as a Post-Doctoral Fellow and a Research Engineer with McMaster University, Hamilton, ON, Canada. Since 1991, he has been a Professor of electrical engineering and computer science with Ohio University, Athens, Greece, and the Director of Embodied Intelligence Laboratories. He has cooperated with the National Institute of Standards and Technology, Gaithersburg, MD, in the area of testing and mixed signal fault diagnosis for eight years. He was a Visiting Professor with the University of Florence, Florence, Italy, and Nanyang Technological University, Singapore. He was a Technical Advisor and Senior Scientist with Magnolia Broadband Inc., Bedminster, NJ, and a Consultant with Magnetek Corporation, Menomonee Falls, WI, and Anteon Corporation, Fairfax, VA, a General Dynamics Company. He was a Visiting Faculty with Wright Laboratories—Advanced Systems Research Group and Redstone Arsenal—U.S. Army Test, Measurement, and Diagnostic Activity. For one year, he held the position of an IPA Fellow with the Automatic Target Recognition Group, Wright Research Laboratories. He was a Visiting Researcher with AT&T Bell Laboratories, Murray Hill, NJ, VLSI Systems Research Group, and Sarnoff Research Laboratories, Mixed Signal VLSI Design Group, Princeton, NJ. His current research interests include embodied machine intelligence, motivated goal driven learning, self-organizing associative spatiotemporal memories, active learning of sensory–motor interactions, and machine consciousness, as well as applications of machine learning to autonomous robots and avatars.

Wooi-Boon Goh (M'92) received the B.Sc. degree (with first class honors) in computer science and electronic engineering from the University of Birmingham, Birmingham, U.K., the M.Phil. degree from the University of Warwick, London, U.K., and the Ph.D. degree from Nanyang Technological University (NTU), Singapore in 1984, 1992, and 2006, respectively.

He is currently an Associate Professor and the Head of the Computing Systems Division, School of Computer Engineering, NTU. Before joining NTU, he was a Senior Engineer and Engineering Section Manager with the Mechanization and Automation Department, Hewlett Packard Singapore, Singapore. His industrial engineering expertise is in the area of developing robot-assisted automation systems. His current research interests include computer vision, human–computer interaction, and multimedia and embedded systems.

Daniel Jachyra (M'10) received the M.S. degree in informatics technologies from the University of Information Technology and Management (UITM), Rzeszow, Poland, in 2007. He is currently pursuing the Ph.D. degree with the same university.

He is a Computational Cognitive Neuroscience Researcher as well as a Lecturer and Secretary with the Department of Applied Information Systems, UITM. His current research interests include mechanisms of motivated behaviors, goal creation, machine learning in autonomous agents, and long-term memory structures.