IN0997 - Redes Neurais

Universidade Federal de Pernambuco

Centro de Informática

Time in self-organizing maps: an overview of models

Tempo nos mapas auto-organizáveis: um panorama dos modelos

Aluizio F. R. Araújo

Pós-graduação em Ciência da Computação

Outline of the Presentation -1

- → **Self-organizing Maps: General Concepts**
- → **Temporal Sequence Processing**
- → **Non-hierarchical Temporal SOM**
- → **Hierarchical Models**
- → **Table with the Covered Models**

Introduction

→ The majority of neural models are concerned with *static mappings*: A given input yields an output that converges to a stable *point attractor*.

→ Real-world data are usually *dynamic*: a particular pattern is usually dependent of its antecedents.

→ Main differences between the processing of the *static patterns* and *dynamic patterns*: To consider the temporal order and correlation between the patterns.

→ Many human and animal tasks involve ability to encode, store, recognize and recall spatiotemporal patterns, one of the most crucial characteristics of any intelligent system.

Introduction

→ Most of the artificial neural network (ANN) models for temporal sequence processing are based on:

→ Multilayer perceptron, trained with a temporal version of gradient-based learning algorithms;

→ Temporal extensions of the Hopfield associative memory model.

→ Both approaches use *associative chaining*: a procedure to consider a temporal sequence as a set of associations between consecutive components:

→ The learning processes consists in building temporal associations between input-output pairs.

Introduction

→ ANN trained through unsupervised learning can deal with sequential patterns: To learn temporal transitions and dependencies from one pattern to another, and to use such learned spatiotemporal features to recognize or recall stored input sequences, and predict future items in a learned sequence.

→ Unsupervised models: show biological plausibility, are governed by principles of self-organization, are conceptually simpler and usually faster than supervised learning.

→ Unsupervised neural networks for temporal processing become more common during the 1990s.

→ The self-organizing map (SOM) proposed by Kohonen is the most chosen model to include spatiotemporal sequence tasks.

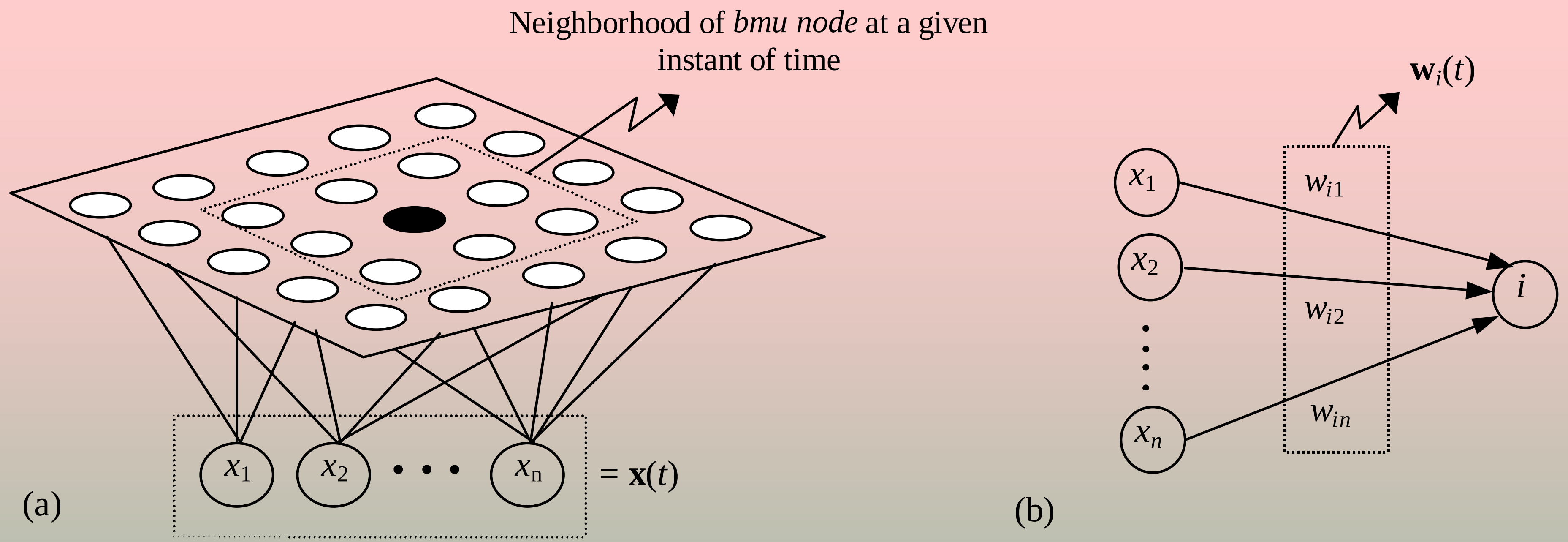
Self-organizing Feature Maps

→ The spatial arrangement of the receptors in the peripheral sensory organs, such as the retina and the skin, is preserved in point-to-point or *topographic connections* in the sensory pathways throughout the central nervous system.

→ Orderly *topographic neural maps* of the visual field are retained at successive processing levels in the brain. Likewise, the surface of the body is represented by a neural map in the somatosensory cortex.

→ Kohonen demonstrated that a simplified version of the more biologically realistic von der Malsburg (1973) model was sufficient to account for some of the major qualitative phenomena associated with the development of topographic cortical maps in real brains.

SOM of Kohonen



(a) Sketch of a typical 2D SOM architecture (b) Weight vector for neuron i in the array.

SOM of Kohonen

The learning algorithm summarized in following steps:

(i) Read a stochastic input vector $\mathbf{x} = [x_1, x_2, ..., x_n]^T \in \Re^n$;

(ii) Determine a winning or best-matching unit (*bmu*) of the map through a similarity measure between the input and weight vectors:

$$bmu = \arg\min_i \{\| \mathbf{x} - \mathbf{w}_i \|\} \qquad (1)$$

where a weight vector of *i* defined as $\mathbf{w}_i = [w_{i1}, w_{i2}, ..., w_{in}]^T \in \Re^n$;

(iii) Update the weights of the winner neuron and its neighbors:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + h_{bmu,i}(t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \qquad (2)$$

where $t = 0, 1, ...$, $\mathbf{w}_i(0)$ are usually random, $h_{bmu,i}(t)$, *the neighborhood function*, is a smoothing kernel defined over the neurons in the array.

SOM of Kohonen

The process stops when the map converges.

To assure convergence: $h_{bmu,i}(t) \to 0$ when $t \to \infty$.

A widely applied neighborhood kernel can be written in terms of the Gaussian function, as follows:

$$h_{bmu,i}(t) = \alpha(t)\exp\left(-\frac{\| r_{bmu} - r_i \|^2}{2\sigma^2(t)}\right) \qquad (3)$$

where $r_{bmu} \in \Re^2$ and $r_i \in \Re^2$, $0 < \alpha(t) < 1$ is the learning rate factor, the parameter $\sigma(t)$ defines the width of the kernel. Both $\alpha(t)$ and $\sigma(t)$ are monotonically decreasing functions of time.

Temporal Sequence Processing: Definitions and Concepts

→ *Spatiotemporal sequence* (or simply, temporal sequence) is a finite set of time-ordered *n*-dimensional feature vectors, also called sequence items, components, or patterns.

→ Representations: A set of characters from a finite alphabet (e.g., **a-e-r-s**) or a single time-indexed character (e.g., {$\mathbf{x}(t\text{-}1)$, $\mathbf{x}(t\text{-}2)$, $\mathbf{x}(t\text{-}3)$, $\mathbf{x}(t\text{-}4)$} or $\mathbf{x}(t_1)\text{-}\mathbf{x}(t_2)\text{-}\mathbf{x}(t_3)\text{-}\mathbf{x}(t_4)$) are common representations for temporal sequences. A particular isolated symbol may represent either a single characteristic or a vector, containing a group of features.

→ Types of *sequence processing*: learning (encoding and storage), classification, recall or prediction of temporal sequences.

Characteristics of Spatiotemporal Signals

→ *Temporal order* involves the relative position of each pattern within a sequence. For example, sequences A-B-C and C-B-A are different, despite they have identical items.

→ *Metrics* entails the duration of a pattern with respect to the others within a sequence.

→ *Density or Sampling* establishes the number of patterns that are considered to represent a sequence. Such a property influences significantly the performance and robustness of ANN models.

→ *Asymmetry* assures that ordered operations often are not reversible.

→ *Temporal context* of a sequence item is defined as the shortest set of previous items that determines unambiguously a given sequence item.

Characteristics of Spatiotemporal Signals

→ *Temporal degree* of a sequence item is the number of items of the temporal context. For example, the temporal context of the last **E** in *R*-*E*-F-E-*R*-*E*-E is *E*-*R*-*E* and its degree is 3.

→ The *context* and *degree* of a sequence is the longest context and the largest degree of its individual items.

→ The signal is *stationary* if its parameters do not vary in time, otherwise, it is *non-stationary*. An engineering problem: *segmentation*, to search regions where the signal properties are reasonably stationary.

→ *Time-Warping* is a signal property related to the duration of an item within a sequence. One may need to align, in time, the features of a test with those of a reference utterance to compare them: one feature string is warped (stretched or compressed in time) to fit the other.

Models of Short-term Memory

→ *Short-term memory* (STM) is a mechanism to retain information on past sequence items to consider the temporal order and/or temporal dependencies between successive input samples. STM functions are:

•*Maintaining a symbol for a short period of time*: An STM model should retain a symbol long enough to perform properly the task at hand. Each symbol is subject to be forgotten due to decay and interference.

•*Maintaining a number of symbols*: According to a representation, there is a maximum number of symbols that a STM can maintain.

•*Coding the temporal order of sequence elements*: An STM should encode the temporal order of items belonging to a given sequence.

•*Coding the length of presentation of each element of a sequence*: STM must code the presentation time of each element of the sequence.

Two Typical STM Models

→ *Tapped delay line*. It comprises the specification of a "time window" over the input sequence, and the corresponding collection and concatenation of successive samples into a single pattern vector of higher dimensionality. In essence, tapped delay lines convert temporal (dynamic) information into spatial (static), so one can use conventional ANNs in dynamic problems. Time dependence of successive samples is captured by the order of the concatenated elements in the input vector. This is an *External STM* in which STM is placed outside of the network

→ *Leaky Integrators Neurons* (LIN): It is understood as a low-pass filter, added to the inputs and/or outputs of the neurons, to function as short-term memories or integrators. LIN models the ability of the neuron to hold its membrane activity over a period of time. This is an *Internal STM* in which STM is placed inside of the network

Still About STM Models

→ **LIN uses the following generic membrane differential equation**:

$$\frac{dP_i(t)}{dt} = \alpha P_i(t) + I_i(t) \qquad (4)$$

where $P_i(t)$ is the membrane potential of neuron i, $\alpha < 0$, and $I_i(t)$ is the input activation, a a function of time.

→ **Additional Concepts of STM Models**

Memory depth: It indicates how far back into the past the STM model stores information relative to the input sequence.

Memory resolution: It refers to the degree of detail to which information concerning individual elements of the input sequence is preserved. The higher the resolution the more accurate is the reconstruction of the sequence elements.

Non-hierarchical Temporal SOMs (TSOMs)

→ TSOM models are formed by a single map; have an external or internal STM mechanism; are often trained as the original SOM

→ **Conventional SOM Applied to Time-Related Tasks**

•The basic idea is to use a series of best-matching units to produce a time trajectory over the topologically ordered SOM in which each best-matching unit represents the state of the process at a specific instant of time.

•SOM is used to form a display of the operational states of the process. The current process state, the *process operating point*, and its history in time can be visualized as a trajectory on the map, making it possible to track the process dynamics. For instance, a process operator may learn to adjust the control variables in such a way that the operating point stays in a desired region on the map.

Self-organizing Maps with External STMs

→ **Operator Maps** - Lampinen & Oja (1989), Walter, Ritter, & Schulten (1990), Principe & Wang (1995), Vesanto (1997).

Applied to time series prediction.

For a finite sequence $\mathbf{X}(t) = \{\mathbf{x}(t), \mathbf{x}(t\text{-}1), ..., \mathbf{x}(t\text{-}n\text{+}1)\}$, in time-series analyses, one can use a local estimate of next input sample:

$\hat{x}_i(t+1) = G_i\left(\mathbf{X}(t)\right)$, for instance $\hat{\mathbf{x}}_i(t+1) = \sum_{j=1}^{n} w_{ij}\mathbf{x}(t-j+1)$

The winner is the minimum prediction error:

$$\|\mathbf{x}(t+1) - \mathbf{x}_{bmu}(t+1)\| = \min_i\{\|\mathbf{x}(t+1) - \hat{\mathbf{x}}_i(t+1)\|\} \qquad (5)$$

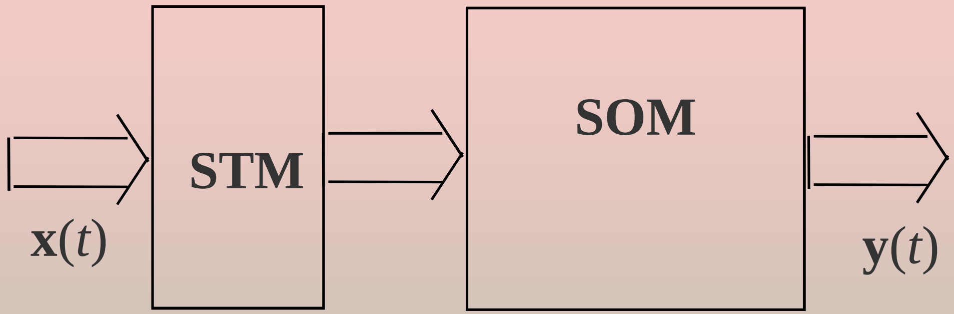
For the neighborhood function, $h_{bmu,i}(t)$, the learning rule is

$$w_{ij}(t+1) = w_{ij}(t) + h_{bmu,i}(t)[\mathbf{x}(t) - \hat{\mathbf{x}}_i(t)]^T \mathbf{x}(t-j+1) \qquad (6)$$

Self-organizing Maps with External STMs

→ **SOM with exponentially weighted decay** (leaky integrator) - Kangas (1990; 1991; & 1994).

Applied to isolated phoneme recognition and segmentation.



Temporal SOM with exponentially weighted decay.

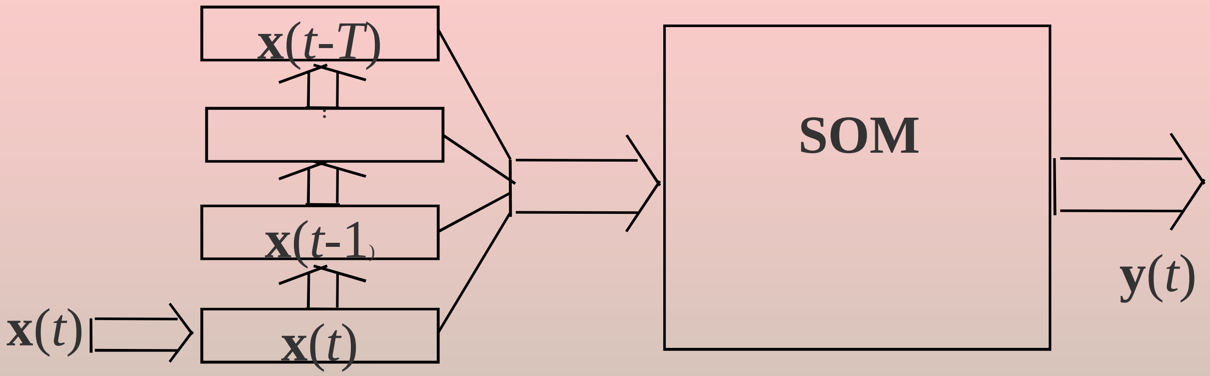
The input to SOM is

$$\bar{\mathbf{x}}(t) = \lambda\,\mathbf{x}(t) + (1-\lambda)\,\bar{\mathbf{x}}(t-1) \qquad (8)$$

Self-organizing Maps with External STMs

→ **SOM with delay line** - Kangas (1990; 1991; & 1994).

Applied to isolated phoneme recognition and segmentation.



Temporal SOM with delay line STM.

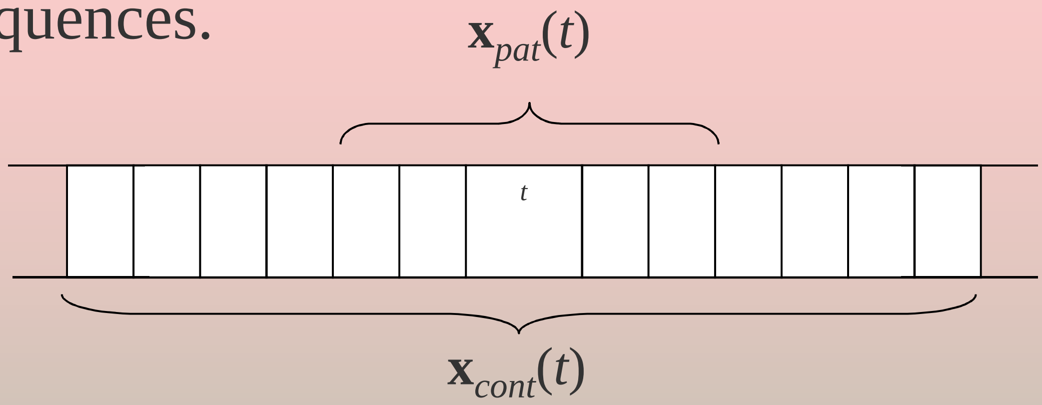
A sequence of *T* input vectors are concatenated by a shift register and then presented to the network.

This strategy increases the computational effort but executes suitably recognition tasks due to the availability of past relevant information.

Self-organizing Maps with External STMs

→ **Hypermap** - Kohonen (1991).

Applied to speech/phoneme recognition and processing of biological sequences.



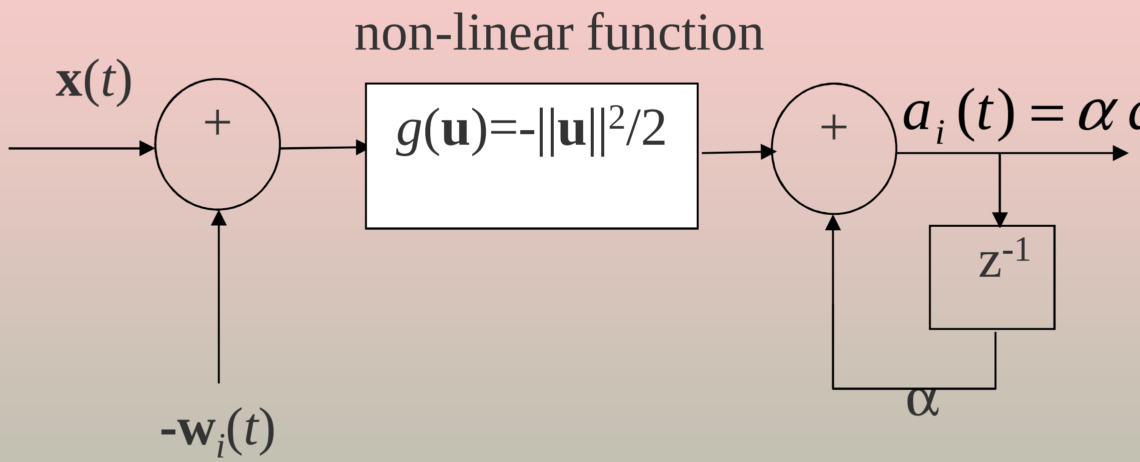
Different time windows centered on a same time step *t* are formed to construct two types of network input vector: a pattern vector $\mathbf{x}_{pat}(t)$ and a wider vector $\mathbf{x}_{cont}(t)$.

Each neuron has one group of inputs from $\mathbf{x}_{pat}(t)$ and other from $\mathbf{x}_{cont}(t)$: The context vector selects a subset of nodes in the network, then the best-matching neuron is then chosen on the basis of the pattern vector.

Self-organizing Maps with Internal STMs

→ **Temporal Kohonen Map (TKM)** - Chappell & Taylor (1993).

Applied to sequence recognition, robot navigation, and time-series analysis.



$$a_i(t) = \alpha\, a_i(t-1) - (1/2)\left\|\mathbf{x}(t) - \mathbf{w}_i(t)\right\|^2$$

where $0 < \alpha < 1$ is the memory depth constant, $\mathbf{x}(t)$ is the input vector, and $\mathbf{w}_i(t)$ is the weight vector of node i. The winning neuron: $a_{bmu}(t) = \max\{a_i(t)\}$ and weights are updated as in the original SOM.

Self-organizing Maps with Internal STMs

→ **Sequential Activation Retention and Decay Network (SARDNET)**
- James & Miikkulainen (1995).

Applied to sequence recognition: sequences of binary and real numbers and phonetic sequences of English words.
SARDNET: A simple TSOM with retention and decay mechanisms to form a unique set of activated neurons for each distinct input sequence.

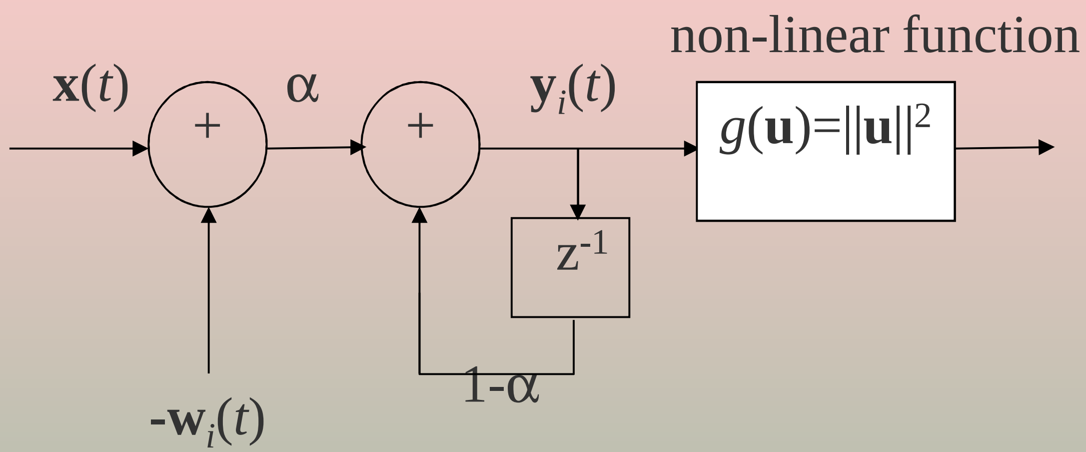
Further leaning steps (considering the two of SOM):
(3) Best-matching units are excluded from subsequent competitions;
(4) Activation rule with an exponentially decaying STM: $a_i(t+1) = \eta a_i(t)$, where $0 < \eta < 1$ is a decay parameter and $a_i(0) = 0$ for all units i.
At the end of a sequence presentation, the strength of neuron activation represents the temporal order (the lowest activation - first item).

Self-organizing Maps with Internal STMs

→ **Recurrent Self-Organizing Map (RSOM)** - Varsta et al. (1997), Koskela et al. (1998a, b).

Applied to time-series prediction and EEG signals.

RSOM: The leaky integrators in TKM are moved from the output to the input units.



$\mathbf{y}_{bmu} = \min_i \|\mathbf{y}_i(t)\|$;

$\mathbf{y}_i(t) = (1 - \alpha)\mathbf{y}_i(t-1) + \alpha[\mathbf{x}(t) - \mathbf{w}_i(t)]$;

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + h_{bmu,i}(t)\mathbf{y}_i(t).$$