Abstract

After the revival of interest in connectionism in the eighties and its successful application to pattern recognition problems, the time has come to consider its role in the field of temporal processing. We present here a general overview of the field of temporal neural networks. In order to give a broad framework to this presentation, we first present general properties of time that are used by AI models. This sets out the properties of time: - on its own, - with respect to a problem, - with respect to a model. We then present a short summary of time processing in symbolic AI. The main part of this article, a classification of temporal neural models, is introduced by a short presentation of basic connectionist models. This classification is then made and several relevant examples are presented. We conclude the article with underlining the difference between temporal reasoning and neural temporal processing, and give an introduction to the following papers of this Sigart special section.

1 Introduction

Since the beginning of the seventies, Artificial Intelligence workers have been designing models aiming at automation of mental processes such as reasoning, understanding language, etc. Some of these capabilities having a temporal dimension, AI models were later adapted to take this dimension into account. A special section of the July 1993 Sigart issue has been devoted to such systems [28].

On the other hand, since the beginning of the eighties, neural network models (re)appeared to take into account sub-symbolic tasks such as pattern recognition. Let us consider the neural network approach as part of the AI field, and call it Connectionist AI (CAI) in opposition to Symbolic AI (SAI). First connectionist models dealt with spatial tasks (without temporal dimension) as for pattern recognition. Later, CAI researchers tried to tackle problems with a minor time component, through adapting standard models as for instance speech understanding and the well known TDNN architecture of Lang & al. [19]. Now there are developments of new connectionist architectures the foundations of which are based on a temporal dimension.

The aim of this Sigart special section is to describe examples of CAI models which are characteristic of main approaches of time modeling with neural networks. The aim of this paper is to introduce major concepts, main neural network models and to set up a framework for comparing SAI and CAI models with respect to time processing.

In section 2, we attempt an investigation of time through properties which are expected to be associated with it, without the influence of a specific problem or model. Section 3 is devoted to a short summary of SAI models which take into account time which is based on the July 1993 Sigart issue on SAI temporal reasoning [28]. In section 4 we sketch a short introduction to the neural network approach and present the main traditional architectures which are referred to in the issue. Section 5 suggests a classification of major neural networks which deal with time. It is followed in section 6 by a brief comparison between SAI and CAI models dealing with time processing. Last section introduces following papers of this issue.

2 Time, what is time?

This paper makes use of major concepts:
- time,
- phenomenon,
- problem,
- model.

So as to make our vocabulary clear, let us begin by giving our understanding of them (without any philosophical pretension).

Before, we must remember that when using these concepts, we assume that there is a human being who observes the phenomenon, specifies the problem, or designs the model.

What is more complicated than trying to define time? The problem of its definition in general deals with many fields (going from philosophy to physics through cognition). We will then use the word "time" with the usual intuitive meaning corresponding to common sense. The next section will however give some basic properties associated with this notion.

A phenomenon is a sequence of observations – let us call them events– corresponding to an entity which evolves in time. For instance a phenomenon can be a sequence of pictures of a ball falling down or a sequence of sounds corresponding to a natural language sentence utterance.

A problem consists in finding a corresponding outcome given some observations of a phenomenon. For instance speech recognition is a typical problem; the input is a signal or a phoneme lattice, the output is a sentence.

A model, is aimed at solving a problem. It is twofold:
- a representation of characteristic aspects of the phenomenon
- a process which handles this representation so as to simulate phenomenon information processing.

2.1 Minimal properties expected from time

Rather than giving a definition of what time is, we want here to exhibit a set of minimal properties this notion has to possess so as to define on what basis the different approaches are grounded.

What first comes to mind is the fact that time is ordered: there is past and future; an event is before or after another event. Moreover, this order is intuitively total, i.e. any two instants can be compared. But the fact that the order is total "in the future" is not as straightforward as it might look at first glance: can we always say between two forthcoming events which one will be the first and which the last? The fact the order has to be total or not will then be left unanswered. Another question raised with orders is the necessity for it to have a "+∞" and/or a "−∞". This question will also be unanswered since time problems usually take place between a beginning time point and an end time point: it does not matter whether time "begins", "ends" or not.

Beside the order, another important fact about time is that we measure it. For instance, "an event happened 10 minutes before another": a duration has a measure. The metrics is then an important property of time that must be listed within the minimal properties.

The next property one can think of is the density of time: between any two moments there is another one. But very often in problems, especially with computers, time is sampled in discrete instants. Thus time is represented as a sequence of discrete time steps rather than as a continuous variable. If not, it is "natural", especially for people with a scientific background, to think time as the set of real numbers as in physics. This aspect of time is also used by some ANN models as we will see. What is important to notice here is that the model incorporates all the preceding properties (order, metrics, density) and moreover provides all the properties of a mathematical parameter with respect to which one can do all mathematical operations allowed (such as derivation for instance).

Another property of time one may think of is the fact that "one cannot go back in time": time is not reversal; it is called "asymmetric". This property is implicitly present in the mind of the designer, but not necessarily in the model.

All the preceding aspects of time are summed up in table 1.

2.2 How is time considered with respect to the problem?

Characterizing time within a problem leads to a set of properties such as those mentioned above.

To build a synthetic point of view on that topic implies a classification which cannot be other than arbitrary (or subjective). We want to emphasize the fact that a classification comes from an observer's point of view on a phenomenon; it is not an intrinsic property of it. We suggest a classification which favors the use of time in respect to the phenomena: time as a property of one phenomenon / time as a relation between several phenomena.

---

Table 1: Some minimal properties expected from "time". Relevant ones for the purpose of this article are marked with '0'.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>order (total?)</td>
<td></td>
</tr>
<tr>
<td>metrics</td>
<td></td>
</tr>
<tr>
<td>density (continuous / discrete)</td>
<td></td>
</tr>
<tr>
<td>set of real numbers</td>
<td>infinite</td>
</tr>
<tr>
<td>asymmetry</td>
<td></td>
</tr>
</tbody>
</table>

2.2.1 Time as a property of one phenomenon

Let us consider some typical examples:
- a speech utterance,
- a moving object,
- an internal combustion engine.

These phenomena are usually considered as time functions. They can be described by a formula such as:

\[ y = f(t) \]

Among these phenomena, different time notions can be found, from simple ones to deeper ones. Simplest ones correspond to a time model limited to order relation. Deeper ones are illustrated by time models based on metrics or even . Let us categorize these differences through a property of \( f(t) \) function.

Consider two time points: \( t_1 \) and \( t_1 + \Delta t \), close to one another (with respect to the time length of the whole phenomenon). For a moving object for instance, \( f(t_1) \) is close to \( f(t_1 + \Delta t) \) as shown in figure 1. We will call it local correlation. This is not the case with speech signal (or phoneme sequence) as shown in figure 2.

![Figure 1: Local correlation phenomena. Example: trajectory of a moving object.](image)

Moreover the combustion engine includes another property: periodicity. This can be described by:

\[ f(t + \tau) = f(t). \]
In that class we may find oscillatory systems or systems which move on a limit cycle. This will be called *global correlation*.

We can then characterize \( f(t) \) in such a way:
- no correlation (neither local nor global); example: speech (especially for phoneme sequence).
- local correlation (no global correlation); example: a moving object.
- global correlation; example: a combustion engine.

To this set of examples we might add the case of watch hands which would be characterized by local and global correlations.

Lastly among these phenomena, we can distinguish those which make use of a continuous or a discrete function. For instance the moving object is represented by a continuous function while a function describing successive macro-states of the combustion engine cycle is discrete. As to speech signal, depending on the the way we observed it, it can be viewed as continuous (signal) or discrete (phoneme sequence).

### 2.2.2 Time as a relation between several phenomena

Consider two temporal phenomena \( f_1 \) and \( f_2 \) which are represented by formulas:

\[
y_1 = f_1(t) \quad y_2 = f_2(t)
\]

For sake of simplicity, we assume \( y_1 \) and \( y_2 \) to be boolean variables (0 or 1).

Besides properties which were introduced for the characterization of one phenomenon, we shall add properties concerning the relation between two phenomena such as:
- precedence,
- simultaneity,
- synchrony.

These concepts being mentioned later, let us give a short description of them corresponding to their use in this paper.

Phenomenon \( \Phi_1 \) is *before* \( \Phi_2 \) if there is a time point \( t_0 \) such as:
- there is a time point \( t_1 < t_0 \) such as \( f_1(t_1) = 1 \)
- \( f_1(t) = 0 \) for \( t > t_0 \)
- there is a time point \( t_2 > t_0 \) such as \( f_2(t_2) = 1 \)

Two phenomena are *simultaneous* at time \( t_0 \) if their state changes at \( t_0 \) and if in time intervals before \( t_0 \) and after \( t_0 \) they are fixed (cf figure 3); interval length depends on temporal scale of the phenomena (and subjectivity of the observer).

This definition may easily be generalized to an observation of the whole time interval.

Two phenomena are *synchronous* within a time interval \([t_1, t_2]\) if there is a time length \( \tau \) such as for each \( t \) between \( t_1 \) and \( t_2 \), \( f_1(t) = f_2(t + \tau) \) (cf figure 4).

In other contexts, some of these concepts could be described with reference to notions such as: period, phase, etc.

### 2.3 Time with respect to the model

After having characterized how the notion of time can be considered with respect to a phenomenon, we now want to focus on how AI models may be related to time. How do they take this notion into account?

First of all, a model is designed for a certain problem or category of problems. So the way a model introduces time is related to the fact that time is or is not present in the problem considered. If not, we will say that the model uses time. Otherwise the model processes time already present in the problem.

In the case where time is used, one can ask for what properties (listed in table 1) it is used and why this new parameter is called "time".

---

*Figure 2: No local or global correlation*

*Figure 3: Simultaneous phenomena*

*Figure 4: Synchronous phenomena*
In the case where time is processed, the relevant questions to be answered are: “How are processed (by which means) the characteristic properties of time present in the problem?”; “Is there some property not present at the problem level but included at the model level?”.

One can also introduce another differentiation between models: do they introduce time as an external dimension of the model or do they fully integrate time in the basic concept of the model? Time will be considered as an external dimension if its processing is due to some preprocessing. It is said to be internal if the model itself contains some way of handling it. Of course, a model cannot use time in an external manner: if time is used (as defined earlier) it is necessarily in an internal way.

A third distinction can be made about time with respect to the model: time handling in the model is either explicit or implicit. When some time parameters are explicitly present in the model (e.g. in the equations), we would say that the model integrates time explicitly. It is implicit in the other cases.

3 Short summary of SAI temporal models

“The French Revolution lasted from 1789 to 1799”
“Louis XVI was executed in 1793”

How to automatically deduce that Louis XVI has been executed during the Revolution?

Let us call the proposition “John eats an apple” (at time $t_1$) $P_1$ and the proposition “The apple has been eaten” (at time $t_2$ after $t_1$) $P_2$. How to automatically deduce that $P_2$ is true at time $t_2$ from the fact that $P_1$ was true at time $t_1$?

To be able to answer such questions automatically, SAI models of time have been designed. A seminal paper on that topic was written in 1983 by J.F. Allen who introduced an interval-based temporal logic with a reasoning algorithm based on constraint propagation [3].

There are two main kinds of time model:

- models based on time points; the semantics of each proposition is defined in respect to a parameter which corresponds to the instant when the event takes place. Example: Louis XVI was executed on January 21st 1793. The time point is the atom of the temporal representation (it might be chosen on an arbitrary scale).

- models based on intervals; the representation atom is the interval; J.F. Allen introduced a set of 13 relations between intervals: before, meets, overlaps, starts, during, finishes, equals, after, ... [3]. Example: the Middle Ages (from $V^{th}$ to $XIV^{th}$ century) is before the Renaissance (from $XV^{th}$ to $XVI^{th}$ century).

Within a formal system such as first order logic, introducing time includes:

- modal temporal operators; example: $F\Phi$ : $\Phi$ is true in some future time

- a temporal precedence relation which may be characterized by properties such as transitivity, linearity, density, etc. which are described through supplementary axioms (see Cocchiarella [9], McDermott [23] and Prior [24]).

It is important to notice that even if formally an interval is a couple of time points and a time point is a null length interval, with a cognitive point of view time points and intervals are different and cannot be reduced to a single concept. Indeed, for instance an interval can be defined by other characteristics than its beginning and ending points. The Renaissance is a period of time characterized by social, cultural events; there isn’t any precise event or date which identifies its beginning or end; only a posteriori it is possible to determine an approximate period of time corresponding to the Renaissance ($XIV^{th}$ and $XVI^{th}$ century). Another example: we might know that storming of the Bastille occurred during the Revolution even without knowing the exact date of this event.

A recent survey of temporal reasoning in SAI by L. Vila can be found in [30].

To give a flavor of temporal models in SAI, let us mention systems which are described in the July 1993 Sigart special section on that topic:

- M. Boddy describes the Time Map Manager system (TMM) for planning and scheduling [7]. The temporal model allows the user to insert constraints between pairs of time points, and then supports queries concerning necessary and possible temporal relations among these time points. This system has been applied to operation planning for the space shuttle, and communication scheduling for plane flight management system.

- A. Gerevini, L. Schubert and S. Schaeffer designed a temporal reasoning system (“Timegraph”) based on instants including metric information processing [12]. Representation relies on partitioned graphs.

- F.A. Barber describes “a metric time-point and duration-based temporal model” [5]. This approach is based on both time points and intervals which are integrated in a common framework. Temporal constraints are represented in a network structure which is processed by a network manager.

4 A short introduction to connectionist models

The aim of this section is to introduce the main concepts of connectionist network field and to describe models which are mentioned in this issue, so as to get the reader familiarized to them.

The word “connectionism” refers to a set of models which are inspired by neural architecture (more precisely by what we believe as being the medium of information processing by the brain). An ANN model is a structure which is composed of a large set of units called “formal neurons” (McCulloch and Pitts [22]) as represented in figure 5.

Each unit is connected to a subset of other units. Each connection is weighted (the weight on the oriented connection
from unit i to unit j is called \( w_{ij} \). Taking into account inputs \( x_i \) coming from upstream units, each unit j (except for network input units) processes a usually non-linear function \( f \) of the input weighted sum:

\[
y_j = f \left( \sum_i w_{ij} x_i \right)
\]

Typically \( f \) may be: \( f(x) = 0 \) if \( x \) is less than some threshold and \( f(x) = 1 \) if \( x \) is greater than this threshold.

The weight set \( \{w_{ij}\} \) is the network memory. It is set up in a learning phase. Algorithms have been designed for that purpose. After training, the neural network can perform a task such as: association, classification involved in pattern recognition, sensorimotor coordination, etc.

Different network structures may be found:

- **Perceptron (cf figure 6)**, introduced by Rosenblatt [27], is composed of an input layer directly connected to an output layer. It can perform tasks which correspond to linearly separable functions.

  ![Figure 6: Perceptron. The cross-lines between input and output represent a full connection: every input unit is connected to every output unit. The architecture presented here provides a character recognition function. The network input is a noisy pattern of the character 'M'; the network output consists in a vector: one unit for each recognized character. The 'H' unit is partially active whereas the 'M' unit is the most active one.](image)

- **Multi-layer Perceptron (MLP) (cf figure 7)** is a Perceptron with one or several unit layers between the input and the output, called "hidden layers"; it can approximate any real function as demonstrated by Funahashi [11]; the Perceptron and MLP are feed-forward networks, i.e. information goes downstream only.

- **Recurrent networks (RN)**.

  There are different kinds of RN:

  - Hopfield's architecture [14]: each unit is connected to each other
  - recurrent MLP: either the hidden layer (Elman [10], figure 8) or the output layer (Jordan [17], figure 9) state at time \( t \) is "copied" in a supplementary input vector for computation at time \( t + 1 \). When computing a new state, the information goes downstream, as in classical MLP; the supplementary input vector brings context information about the past. This architecture is able to memorize and recall sequences (example: poems, robot arm positions).

  ![Figure 7: Multi-layer Perceptron.](image)

  ![Figure 8: Recurrent network: Elman's architecture](image)

  Rohwer's paper in the present issue gives several examples of recurrent architectures.

Remark: the architectures we described are aimed at memorizing and recalling associations. This is one main function a neural network can perform. Other functions are also provided by other ANN models. Main ones are:

- classification (models of Kohonen, Grossberg)
- optimization: search for a function optimum (Hopfield's model).

5 Time in CAI

As already mentioned in section 2.3, time can be introduced in ANN models either in an external or in an internal man-
In the first subsection, we will present a step by step integration of time into the ANN models which will lead to a classification of these models which is summed up in figure 10. No example will be given in this first part. This is left for the second subsection where several temporal ANN models are presented with respect to the analysis made in the first one.

5.1 Classification

In the integration of time into ANN models, the first step is not to introduce it at all but to leave time outside the ANN model and to preprocess the data so that standard (without time) ANN models can proceed with the task. Time is there only preprocessed in a time to space transformation, and the network accesses only spatial information which contains a dimension whose semantics is related to time.

The actual introduction of time in a neural model can be made at several levels. First of all, time can be used as an index of network states. It is only used to keep the preceding state of nodes (to be reintroduced at the following step). Such time (state index) is a mechanism of the ANN but is not necessarily used to handle a time problem. We may say that time is implicitly present in the model. Order is the only property of time used here: a state occurring previously is reintroduced into current step processing.

A step further in the introduction of time in a neural model is to represent it explicitly at the level of the architecture (the network) by some delays of propagation (time weights) on the connections. In that case, the metric aspect of time is fully used.

The deepest integration of time in neural models is reached when introducing this notion at the level of the neuron itself. There are several ways of doing this, as more as this motivation for including time in ANN is at its very beginning. We will present here the main ideas underlying this approach.

First, we want to underline the fact that introducing time at the neuron level or at the connection level are not equivalent. Qualitatively, introducing time at the connection level with some time weights allows a global process (at the network level) of time. Output of the network can then depend on a given time slice of network inputs whose length depends directly on the network design and the time weights. On the other hand, time at the neuron level introduces a local processing of time and allows some temporal robustness: two inputs of a given neuron do not need to be exactly simultaneous to activate this neuron, one can be slightly later than the other.

Introducing time at the neuron level can be done from two different points of view: either by keeping biological properties and considering the biological plausibility of the model, or by building up a model from an engineering point of view, introducing time ignoring biology.

The first way usually leads to more or less complicated differential equation models of the neurons. This approach is often referred to as "integrate and fire" model. The principle consists in summing the neuron inputs over a period of time and when this sum becomes greater than a given threshold (for each neuron) to make the neuron fire (i.e. change state).

The engineering way can be either purely algebraic as done by Vaucher [29] or can consist of introducing an artificially time varying bias or even considering that equation (1) concerns equilibrium and can be generalized by

$$\frac{dy_j}{dt} = -y_j + f \left( \sum_i w_{ij} x_i \right)$$

The introduction of continuous time often leads to dynamical properties. The study of dynamical systems is in general not simple and leads to complex behavior implying oscillations and synchronization (see for instance Hirch [13], Ramacher [25] or Lumer [20]).
5.2 Some temporal ANN models

Time in neural networks is used for several different properties (with respect to section 2.1) in many different kinds of problem (with respect to section 2.2) and is implemented at different levels in the model as shown previously. We present here some neural models recently developed, with respect to those three points.

Concerning the first kind of integration of time dimension, as an external one, the typical model is the well known TDNN of Lang, Waibel and Hinton [19]. The model is a modification of the MLP whose input consists of spectrogram “slices” which are delayed through a delay line. The network input at a given time step is the whole delay line at this time step. This set of time slices, which is shifted each time step, consists of the calculation of the hidden nodes. In order to keep the time consistency, the weights are set to be equal among time slices. The problem considered in [19] is to recognize four kinds of phoneme in a sequence of spectrogram slices.

Another technique consisting of modifying already existing architectures (MLP) was presented in RN architectures in section 4. The kind of problem handled with those architectures is the learning of a sequence.

In both TDNN and RN architectures, time is only used for its order property: what is important is the sequence.

The works of Béroule [6], Jacquemin [16] and Amit [4] are three different but representative approaches to connections carrying temporal information: units are not only connected through weighted connections but also through delays (“temporal weights”). The time metrics is explicitly represented at the network level. Problems that are tackled with such architectures are typically temporal matching between events. In Amit’s work [4], there are even two levels in the architecture: the classical one (no delay) which recognizes shapes, and the delayed one which is able to count the recognized shapes.

Klaassen [18] too manages connections with temporal weights, but he also introduces time at the neuron level. The neuron model, coming from biology, consists of modeling neuron electrical activity propagation with cable differential equations (like electrical cables). This involves many properties for the neuron unit such as refractory period: the neuron is “deaf” to its inputs for a period. Time is used here as a mathematical parameter and has thus all the properties of a real number.

Other examples of temporal neuron models are biological approaches presented by Abbott and Kepler [1] or Rinzel and Ermentrout [26]. The purpose is there to introduce time into an artificial neuron model as close as possible to biological observations. Those models lead to finite-difference neural equation. They can be included in any classical architecture to provide ANN with temporal properties and are a good basis for temporal neuron models even for engineering problems.

But time also enters the ANN world from the other side: as an artificial tool to solve a problem rather than as a biological property. The problem to be solved is the so called “binding problem” which consists of representing links between different items through their identical time coordinates. This approach first described in 1981 by Von der Malsburg [31] is well represented by the works of Aijanagadde and Shastri [2] or Martin [21]. Time is used here as a new dimension in order to solve a problem (binding) whose spatial solution would be too large to be implemented (too many combinations). The property of time used here is nothing else but a new dimension. The binding is done through oscillations of neural activities. For instance, to represent the fact that car_1 is red, the activities of neurons representing car_1 and red will be simultaneous (see figure 12).

Oscillations are also used in the representative work of Horn and Usher [15]. Memorized shapes have an oscillatory behavior so that several different memorized items can then be represented at the same time even if they share the same set of neurons: they oscillate with different phases. Time dimension is here introduced at the neuron level as a mathematical parameter for (temporal-)parametric equations of the neuron activity. The approach chosen is a time-varying bias. This kind of oscillatory neuron described by coupled differential equations (the activity and the bias) can be justified by biological approaches [1, 26].

Our approach [8] consists also of integrating time dimension at both neuron and connection levels. But we do not model the neuron with cable equation as in [18], which can be very long to integrate for large numbers of neurons. Our model is built on a simpler neuron model such as those described in [1] or [26]. The main characteristic of our approach is to use both spatial and temporal metrics in order to simplify input processing. The kind of problem we are interested in is time and space correlation detection such as for example motion recognition. Time is considered here much more as a meaningful dimension on its own rather than a useful parameter.

Figure 11: TDNN architecture: a MLP architecture modified so that units take their inputs among only a part of the previous layer corresponding to a time slice. Different time slices are represented with different line styles.

Figure 12: Example of binding through synchronous activations: car_1 is bound to red, green and car_2 are bound to nothing.
Table 3: Several SAI and CAI models with respect to points developed in section 2. $\Phi$ means that the characteristic is not relevant for the model.

6 Differences between SAI and CAI models

In table 3, we sum up the properties of both SAI and CAI approaches with respect to time properties mentioned in section 2.

A first difference between SAI and CAI models comes directly from the intrinsic scope of each field. SAI models take into account only symbolic features while CAI models process mainly sub-symbolic informations (pixels or frequencies for instance). With respect to time, this difference leads to the fact that time scale is different. In SAI models, time unit is hour, day, or year; time processing consists of reasoning about events, i.e. deducing event relations from their time features. Thus in the SAI field, time is mainly considered as an order relation; some models however include metrics (sensitivity to time distance). In CAI models, time unit is below minute, even below second; time processing consists of building a synthetic information from temporal correlations.

Another important difference between SAI and CAI models which is a consequence of the fact that the problem domains are different, concerns the data processing. In SAI models, time processing consists in reasoning on temporal data and deducing new relations between events. In CAI models, time processing mainly consists in using temporal correlation to store, recall and combine information.

In table 3, we can see that for all SAI models, time modeling is characterized by: processed, internal, explicit, which can be considered as the aimed properties for a model. Time has been integrated in the basic concepts of the model. On the other hand, some CAI models (out of Amit’s and Béroule’s models) are limited to using time, or introduce it as an external dimension, or manage it in an implicit way. This state of affairs can be ascribed to the childhood of the field of time processing in CAI.

7 Introduction to the following papers

We gave in this article an overview of what is presently done in the field of temporal neural networks. The aim was not to be exhaustive but rather to exhibit the main ideas and realizations in this area. The following papers of this issue provide other points of views as well as concrete examples.

The first article, written by C. Jacquemin, shows an example of both classical and time weighted connections. It is applied to natural language disambiguation.

The next paper, written by T. Catfolis, begins with another general point of view on time in AI. It then describes an example of cooperation of a recurrent network and a feedforward network. This model is applied to a valve diagnosis problem.

Two examples of temporal neural networks applied to speech recognition are finally given. M. H. Nguyen and G. W. Cottrell show the application of a recurrent network to a voicing distinction task. Their neuron model is dynamical (a differential equation is integrated into finite-difference equation).

As a concluding article, R. Rohwer presents another overview of the domain of temporal ANN. He insists on the relationship between feed-forward and recurrent networks and gives several examples of speech recognition. He concludes by providing future directions.
Acknowledgments

We want to thank the Sigart’s Editors-in-Chief D. Powers and L. Hoebel for their invitation, the authors who contributed to this special section, P. Weyerbrown for his help in improving presentation of this paper and C. Jacquemin for relevant comments.

References


