The RRBF
Dynamic representation of time in Radial Basis Function Network

Zemouri Ryad, Racoceanu Daniel, Zerhouni Noureddine
Laboratoire d'Automatique de Besançon
Institut de productique
25, rue Alain Savary – 25000 Besançon
France

Abstract - This paper introduces the Recurrent Radial Basis Function networks (RRBF) for recognition of simple temporal sequence. The RRBF combines features from recurrent neural network and Radial Basis Function networks (RBF). An application has been developed by implementation on the IBM/ZISC (Zero Instruction Set Computer) card for temporal sequences recognition.

I. INTRODUCTION

Some applications of Artificial Neural Networks, like monitoring and safety, require architectures able to treat the temporal aspects. One of the first architectures of temporal neuronal networks (TDNN - time delay neural network) has been introduced by Waibel & al [11] for phonemes recognition. Long training times and multiple parameters need careful adjustments and make this network difficult to apply to new tasks. Another type of temporal neuronal networks (TDRBF - time delay radial basis function) with the same representation (spatial) of the time has been used for the same application by Berthold [3]. This type of network combines features from the spatial representation of time of the Multi Layer Perceptron (TDNN) and the RBF networks. His major advantage is the simplicity of the training process (characteristic of the RBF network). The only drawback of such a network is due to the spatial representation of the time. The desire to combine the advantages of Radial Basis Function (RBF) with the dynamic representation of time led to the Recurrent Radial Basis Function (RRBF) presented in this paper.

The section II of the paper discusses the different representations of the time in neural networks [2]. After a brief description of the RBF networks, we present the architecture of the RRBF networks. We will conclude by the results of the learning of temporal sequences by implementation on the ZISC card (Zero Instruction Set Computer) of IBM.

II. REPRESENTATION OF TIME IN NEURAL NETWORKS

The representation of time given by [6], [5] makes appear two types of solutions (Fig.1). The time in neural networks can be represented by an external or an internal mechanism. These two terms correspond respectively to a spatial and dynamic representation of the time [8].

In the spatial representation, the time is introduced in the model with the help of an external mechanism. The objective is to find particular architectures permitting to perform dynamic parameters. The known networks using actually this concept are: TDNN (Time Delay Neural Network) used with the famous vocal recognition software NetTalk, and the TDRBF (Time Delay Radial Basis Function) used for the phonemes recognition [3]. The major inconvenience of these algorithms is the existence of an external interface with the environment to delay and store data. The second disadvantage is the use of a temporal window that imposes a limit of the sequence length. The major advantage of the TDRBF networks compare to the TDNN is the training flexibility, and the reduce number of parameters to adjust the training time [3].

Concerning the dynamic representation of time, when the time is an internal mechanism, the time can be implicit (recurrent networks) or explicit (time variable appears on the network connections or neurons). For explicit time at the neuron level (the most “biological” approach), we evidentiate methods using algebraic or biologic models. Both methods use very complicate algorithms and ask
The RBF network defined above is incapable to treat dynamical data (temporal inputs). We present in the next section a new architecture of RBF network which can detect features in time.

IV. RECURRENT RADIAL BASIS FUNCTION NETWORK (RRBF)

A. Architecture of the network

The RRBF neural network (Recurrent Radial Basis Function) that we propose uses an internal representation of the time. This property obtained with a self-connection of the input layer neurons procures to the RBF network a dynamic aspect with the simplicity of the training process (Fig.3). We tested with success the RRBF network on an IBM/ZISC card [7] for the training of temporal sequences.

This self-connection has been used on a Multi Layer Perceptron by Bernauer & al [1]. The major inconvenience of this neural network is the complexity of his training process (Back Propagation Algorithm). Indeed, the parameters adjustment is very delicate and requires several tests and a good knowledge of the problem to solve. The flexibility of the training process of the RRBF network (same training algorithm - RCE [10] - as the RBF networks) represents an important advantage of our architecture.

B. Effect of the self-connection

Every neuron of the input layer make a summation at the instant \( t \) between his entry \( I_i \) and his output of the previous instant \( (t-1) \) weighted by the weight of the self-connection \( w_{ii} \). He gives in output the result of the activation function:

\[
a_i(t) = w_{ii} x_i(t-1) + I_i
\]

\[
x_i(t) = f(a_i(t))
\]  

(2)  

```latex
\[
f_i(x) = e^{-\frac{d_i(x)^2}{\sigma_i}}
\]

\[
\text{with } d_i(x) = ||X - C_i|| \text{ measuring the distance between the input vector } X \text{ and the prototype } C_i, \text{ and } \sigma_i \text{ is the size of the influence field (standard deviation).}
\]  

```
where \( a_i(t) \) and \( x_i(t) \) are respectively the activation and the output of the neuron \( i \) at time \( t \), \( f \) represents the activation function of the neurone \( i \) having the expression of the sigmoid below:

\[
f(x) = \frac{1 - \exp(-kx)}{1 + \exp(-kx)}
\]

(3)

and \( w_{ij} \) is the weight of the self-connection of the \( i \) neuron.

To study the effect of the self-connection of every neuron, we consider the input \( I_i = 0 \) and the neuron output \( x_i(t) = 1 \). The neuron will evolve thus, without the influence of the external input \( (I_i = 0) \) [1,2]. The evolution of the output neuron is:

\[
x_i(t) = \frac{1 - \exp(-kw_{ij}x(t-1))}{1 + \exp(-kw_{ij}x(t-1))}
\]

(4)

The diagram of the figure 4 shows the evolution of the output of the neuron in time. This evolution depends on the gradient of \( \Delta \) (inverse of the connection weight of the neurone \( w_{ij} \) ) and also on the value of the \( k \) parameter of the activation function.

C. Learning temporal sequences

The self-connection of every entry neuron procures a certain memory to neurons. This characteristic allows them to take into account the previous entries and not only entries at the instant \( t \). Every \( I_i \) entry represents the occurrence of an event \( i \) of a sequence:

\[
I_i = 1 \text{ if the event } i \text{ of the sequence occurs,}
\]

\[
I_i = 0 \text{ else.}
\]

The algorithm of training used for the RRBF network is the RCE [10] which introduces a new prototype when it is necessary and adjusts the influence field of existing prototype in order to avoid conflicts. The neuronal hardware IBM/ZISC uses the RCE algorithm for the phase of training [7]. This algorithm of training is much easier to process than the back propagation algorithm used by Bernauer & al [1]. Otherwise, the problem of over-training met in back propagation algorithm doesn’t have an effect in the RCE algorithm.

V. APPLICATION

We test the RRBF network on a temporal sequence recognition using the IBM/ZISC neural hardware. This neural card is an implementation of the RBF network describes in section III. It contains until 576 neurons in the hidden (second) layer and it is capable to evaluate more than 200 000 entries per second [7]. In order to endow this card with the dynamic aspect, we modify features of the input neurons layer by soft programming. The obtained network is therefore similar to the RRBF network of the figure 3.

Tested temporal sequences are composed of four independent events (A,B,C,D). During the training process, events are presented to the network one by one and the category is defined after the last event of the sequence is presented to the network (event D for the case of the ABCD sequence). The number of neurons of the entry layer is four (length of the sequence). Otherwise, the number of the two other layer neurons depends on the number of sequences learned by the network. Every neuron of the hidden layer memorizes a prototype (vector of the sequence), and every neuron of the output layer represents a category (sequence). The only parameters that we had to define are weights of self-connections of input neurons \( (w_{ij} = 0.9) \) and the maximal distance of influence fields of neurons of the hidden layer \( (\sigma_i = 10) \). About ten sequences have been learned by the networks (ABCD, ABDC, ACBD, ...). After the training phase, that took only one cycle, all learned sequences are recognized with success, and the phenomenon of the over-training met in the back propagation algorithm doesn’t have an effect on this network.
VI. CONCLUSION

In this paper we presented the RRBF network (Recurrent Radial Basis Function) for the temporal sequence recognition. The major advantage of this network type is the training process flexibility: relatively short training time and few parameters to optimise. The RRBF network is inspired by advantages of the RBF networks and the recurrent networks (Multi Layer Perceptron). The application of this network on the IBM/ZISC card has been achieved with success. Our future works will be centred on the relative mathematical development to the application of the RRBF network on complex problems as the industrial system surveillance (fault detection, localization and diagnosis) where the time plays a very important role in the system evolution.

VII. REFERENCES


