Neurocomputing 72 (2009) 945–955

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Single-layered complex-valued neural network for real-valued classification problems

Md. Faijul Amin [a], Kazuyuki Murase [a,b,*]

[a] Department of Human and Artificial Intelligence Systems, Graduate School of Engineering, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan
[b] Research and Education Program for Life Science, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan

ARTICLE INFO

Article history:
Received 1 June 2007
Received in revised form
25 April 2008
Accepted 29 April 2008
Communicated by T. Heskes
Available online 9 May 2008

Keywords:
Activation function
Classification
Complex-valued neural networks
Generalization
Phase-encoding

ABSTRACT

This paper presents a model of complex-valued neuron (CVN) for real-valued classification problems, introducing two new activation functions. In this CVN model, each real-valued input is encoded into a phase between 0 and π of a complex number of unity magnitude, and multiplied by a complex-valued weight. The weighted sum of inputs is then fed to an activation function. Both the proposed activation functions map complex values into real values, and their role is to divide the net-input (weighted sum) space into multiple regions representing the classes of input patterns. Gradient-based learning rules are derived for each of the activation functions. The ability of such CVN is discussed and tested with two-class problems, such as two- and three-input Boolean problems, and the symmetry detection in binary sequences. We show here that the CVN with both activation functions can form proper boundaries for these linear and nonlinear problems. For solving n-class problems, a complex-valued neural network (CVNN) consisting of n CVNs is also studied. We defined the one exhibiting the largest output among all the neurons as representing the output class. We tested such single-layered CVNNs on several real world benchmark problems. The results show that the classification ability of single-layered CVNN on unseen data is comparable to the conventional real-valued neural network (RVNN) having one hidden layer. Moreover, convergence of the CVNN is much faster than that of the RVNN in most cases.

1. Introduction

Complex numbers are used to express real-world phenomena like signal amplitude and phase, and to analyze various mathematical and geometrical relationships. In order to directly process complex values by artificial neural networks, the complex-valued neural network (CVNN) as well as the complex back-propagation (CBP) algorithm have been developed [3,7,8,10,16]. The properties of the CVNN and CBP have been studied [5,11], and the CVNN is shown to be powerful in applications such as adaptive radar image processing [15], and optical image processing [2,4]. Further extension to multidimensional values has been attempted as well [13]. Some researchers recently have also applied CVNN on real-valued classification problems [1,9].

We are aware of two approaches for the application of CVNN to real-valued classification problems. In Ref. [9], each real-valued input is phase encoded between 0 and $\pi/2$ of unity magnitude complex number, so that the complex-valued neuron (CVN) can receive complex-valued inputs. The role of CVN is two-fold, aggregation and threshold operations. The former role is to aggregate the inputs multiplied by the connection weights, and the latter is to determine the class label by using an activation function. They showed that the CVN is successful in classifying all two-input Boolean functions, and 245 among 256 three-input Boolean functions. The learning algorithm, however, includes a reciprocal of partial derivatives. When the partial derivatives approach zero and the reciprocals become very large, the learning process may become unstable, especially when real world problems are evaluated. In a recent work [1], a multilayer feed-forward architecture of multivalued neuron is proposed. The model encodes real-valued inputs by phases between 0 and 2π of unity magnitude complex numbers, and determines the class label by the complex-valued output, based on the output's vicinity to the roots of unity. It has been shown that the model was able to solve the parity n and two spirals problems, and could perform better in "sonar" benchmark and the Mackey–Glass time series prediction problems.

In this paper, we propose two activation functions that map complex values to real-valued outputs. The role of the activation functions is to divide the net-input (sum of weighted inputs)

* Corresponding author at: Department of Human and Artificial Intelligence Systems, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan.
Tel.: +81 776 27 8774; fax: +81 776 27 8420.
E-mail address: murase@synaspe.his.fukui-u.ac.jp (K. Murase).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.04.006

space into multiple regions representing the classes. Since the net-input of a CVN is a complex number, it is a two-dimensional space. Both the proposed activation functions are differentiable with respect to real and imaginary parts of the net-input. As a result, a gradient-based learning rule can be derived. To present complex-valued inputs to the CVN, real-valued inputs are phase encoded between 0 and π. We will show in the following sections that such a CVN is able to solve linear and nonlinear classification problems such as two-input Boolean functions, 253 among 256 three-input Boolean functions, and symmetry detection in binary sequences.

To solve n-class problems, we considered a single-layered CVNN (without hidden layer) consisting of n CVNs described above, and formulated the learning and classification scheme. The single-layered CVNN has been applied and tested on various real world benchmark problems. Experimental results showed that the generalization ability of the single-layered CVNN is comparable to the conventional two-layered (with one hidden layer in addition to input and output layer) real-valued neural network (RVNN). It is noteworthy that in the proposed single-layered CVNN, the architecture selection problem does not exist. In the multilayered RVNNs which are considered as universal approximators [6], in contrast, the architecture selection is crucial to achieve better generalization and faster learning abilities.

The remainder of the paper is organized as follows. In Section 2, we discuss the model of CVN along with the proposed activation functions. In Section 3, we develop a gradient-based learning rule for training the CVN. In Section 4, we discuss the ability of the single CVN for some binary-valued classification problems. In Section 5, performances of the single-layered CVNN consisting of multiple CVNs are compared to those of the conventional two-layered RVNN on some real world benchmark problems. Finally, Section 6 gives a discussion and concluding remarks.

2. Complex-valued neuron (CVN) model

Since the CVN processes complex-valued information, it is necessary to map real input values to complex values in order to solve real-valued classification problems. After such mapping, the neuron processes information in a way similar to the conventional neuron model except that all the parameters and variables are complex-valued, and computations are performed according to complex algebra. As illustrated in Fig. 1, the neuron, therefore, first sums up the weighted complex-valued inputs and the threshold value to represent its internal state for the given input pattern, and then the weighted sum is fed to an activation function which maps the internal state (complex-valued weighted sum) to a real value. Here, the activation function combines the real and imaginary parts of the weighted sum.

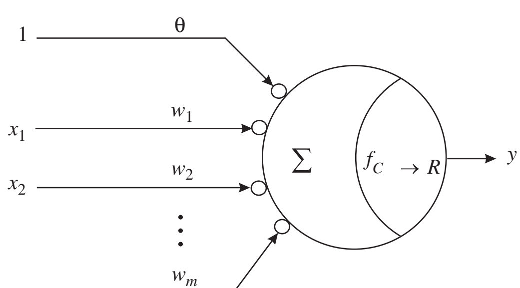

Fig. 1. Model of a complex neuron. The sign $\sum$ sums up the weighted inputs $w_j x_j (1 \leqslant j \leqslant m)$ and the bias θ. $f_{C\to R}$ is an activation function that maps the complex-valued internal state to a real-valued output y.

2.1. Phase encoding of the inputs

This section explains how the real-valued information is presented to a CVN. Consider (X, c) as an input example, where $X \in R^m$ represents the vector for m attributes of the example, and $c \in \{0, 1\}$ denotes the class of the input pattern. We need a mapping $R^m \to C^m$ to process the information with the CVN. One such mapping for each element of the vector X can be done by the following transformation:

$$\text{Let } x \in [a, b], \quad \text{where} \quad a, b \in R, \quad \text{then } \varphi = \frac{\pi(x-a)}{(b-a)} \tag{1}$$

and

$$z = e^{i\varphi} = \cos\varphi + i \sin\varphi \tag{2}$$

Here $i = \sqrt{-1}$. Eq. (1) is a linear transformation which maps $x \in [a, b]$ to $\varphi \in [0, \pi]$. Then by Euler's formula, as given by Eq. (2), a complex value z is obtained. When a real variable moves in the interval of $[a, b]$, the above transformation will move the complex variable z over the upper half of a unit circle. As shown in Fig. 2, the variation on a real line is thus now the variation of phase φ over the unit circle.

Some facts about the transformation are worth noting. Firstly, the transformation retains relational property. For example, when two real values x_1 and x_2 hold a relation $x_1 \leqslant x_2$, the corresponding complex values have the same property in their phases as such, $phase(z_1) \leqslant phase(z_2)$. Secondly, the spatial relationship among the real values is also retained. For example, two real values x_1 and x_2 are farthest from each other when $x_1 = a$ and $x_2 = b$. The transformed complex values z_1 and z_2 are also farthest from each other as shown in Fig. 2. Thirdly, the interval $[0, \pi]$ is better than the interval $[0, 2\pi]$ as we loose the spatial relationship among the variables in the latter. For example, $x_1 = a$ and $x_2 = b$ will be mapped to the same complex value since $e^{i0} = e^{i2\pi}$. It is worth noting that interval $[0, \pi/2]$ can also be used for phase encoding. However, experiments presented in Section 5 and Table 9 show that learning convergence is faster in the case of the interval $[0, \pi]$. Finally, the transformation can be regarded as a preprocessing step. The preprocessing is commonly used even in RVNNs in order to map input values into a specified range, and so on. The transformation in the proposed CVN, therefore, does not increase any additional stage for the process with neural networks. In fact, the above transformation does not loose any information from the real values; rather it lets a CVN process the information in a more powerful way.

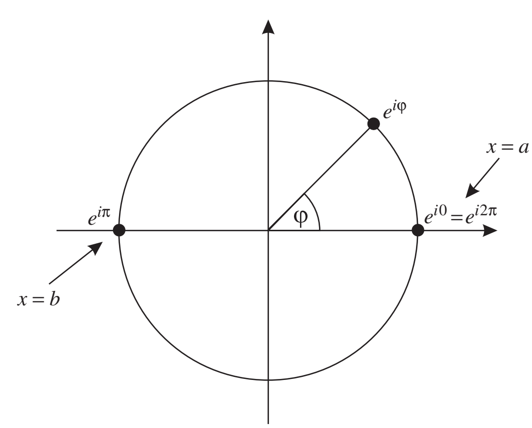

Fig. 2. Phase encoded inputs. When a real value x moves in the interval $[a, b]$, corresponding complex value moves over the upper half of unit circle. $x = a$, and $x = b$ are mapped by e^{i0}, and $e^{i\pi}$, respectively.