Neurocomputing 72 (2009) 2227–2234

Contents lists available at ScienceDirect

Neurocomputing

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

Ensemble of single-layered complex-valued neural networks for classification tasks

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

[a] Department of Human and Artificial Intelligence Systems, Graduate School of Engineering, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan
[b] Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh
[c] Research and Education Program for Life Science, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan

ARTICLE INFO

Article history:
Received 10 July 2008
Received in revised form
5 November 2008
Accepted 29 December 2008
Communicated by T. Heskes
Available online 20 February 2009

Keywords:
Activation function
Classification
Complex-valued neural network
Ensemble

ABSTRACT

This paper presents ensemble approaches in single-layered complex-valued neural network (CVNN) to solve real-valued classification problems. Each component CVNN of an ensemble uses a recently proposed activation function for its complex-valued neurons (CVNs). A gradient-descent based learning algorithm was used to train the component CVNNs. We applied two ensemble methods, negative correlation learning and *bagging*, to create the ensembles. Experimental results on a number of real-world benchmark problems showed a substantial performance improvement of the ensembles over the individual single-layered CVNN classifiers. Furthermore, the generalization performances were nearly equivalent to those obtained by the ensembles of real-valued multilayer neural networks.

1. Introduction

Complex numbers are inevitable from both the theoretical and application perspectives. In order to process such information by artificial neural networks, researchers have developed various complex-valued network (CVNN) models, such as feed-forward and recurrent CVNNs [11,12,21], complex-valued self-organizing map [6], and complex-valued associative memories [14,20]. Recent developments are compiled in [7]. It is very natural that the CVNN would find its applications on the areas, such as telecommunications, speech recognition, image processing, and others, where data to be processed are complex-valued. However, some researchers recently have also applied CVNN to real-valued classification problems by representing and solving the problems in the complex domain.

Researchers have investigated and found that a complex-valued neuron (CVN) could achieve better classification ability than a real-valued neuron (RVN). One of the earlier works [19] studied the discrimination ability of a complex perceptron on Boolean functions up to four inputs. It was shown that the complex perceptron could achieve twice the discrimination ability of a real perceptron. In [22], the real-valued inputs and the class labels were encoded by complex numbers, and then were processed by a CVN to solve the XOR and the symmetry detection problem. In [17], real-valued inputs were encoded by the phases of unity magnitude complex numbers. Depending on the magnitude of the CVN's output, the class label of an input pattern was determined. The CVN could achieve an improvement of 135% over an RVN for the three-input Boolean functions.

There has been another recent approach that used multilayer feed-forward architecture of multi-valued neurons [1]. The approach also encoded the inputs by the phases of unity magnitude complex numbers, but the class labels were encoded by the roots of unity in the complex plane. They showed that their feed-forward multilayer network could successfully solve the parity n ($2 \leqslant n \leqslant 9$) problem and the two spirals problem, and could perform better in the "sonar" benchmark and the Mackey–Glass time series prediction problems.

Most of the aforementioned approaches, however, have some shortcomings. The CVN models of [19] and [22], for example, require careful settings of the target outputs ($\{0, 1\}$ in [19] and $\{1, 0, 1+i, i\}$ in [22]) for different classes. Choosing an arbitrary output encoding scheme ($0 =$ class A and $1 =$ class B, or the reverse setting) will not work for some problems since it was reported in [19] that the CVN could not realize several three-input Boolean functions, while the complementary functions were realized. Assigning output values to different classes is even more complicated for the CVN model of [22]. The same problem exists in [17]. Moreover, the learning algorithm of the CVN [17] may

* Corresponding author at: Department of Human and Artificial Intelligence Systems, Graduate School of Engineering, 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 © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.12.028

suffer from instability due to a reciprocal of derivative, namely when the derivative approaches to zero.

In order to minimize the shortcomings, we proposed a new class of activation functions [2], whose role is similar to the conventional RVN in the classification tasks. The functions combine the real and imaginary part of the complex numbers, and map complex values into bounded real values. Due to the differentiability of the activation functions, a gradient-descent learning algorithm can be easily derived (see [2] for the learning algorithm). We showed that a CVN with these activation functions could successfully solve several Boolean classification problems (including linear and nonlinear problems). We further studied the generalization ability of single-layered CVNN on several real-world multiclass problems, and showed that the performance was comparable to that of the multilayer real-valued neural networks (RVNNs).

It should be mentioned that a CVN or a single-layered CVNN (only one layer of computing neurons, each representing one class) cannot match all possible problems' complexity because of the fixed structure (determined by the number of inputs and number of classes). To improve the performance, one possibility is to use multiple layers of neurons as was done in [1]. However, it is well known that an ensemble of classifiers can achieve better classification ability than that of an individual classifier, provided that the individual classifiers do not make error on the same part of the data [24].

In this study, therefore, we investigate the ensemble methods in the single-layered CVNNs that were developed in our earlier work [2]. Among the ensemble creation methods, we examined two methods, negative correlation learning (NCL) [15] and *bagging* [3]. The former is an explicit method, while the latter is an implicit method [4].

We show here that the ensemble methods can enhance the performance of single-layered CVNNs to a considerable extent. Furthermore, experimental results on various real-world benchmark problems show a comparable generalization performance of the single-layered CVNN ensembles to that of the multilayer RVNN ensembles.

It is noteworthy that any ensemble methods can be easily applied to our CVN model due to its gradient based learning rule. For example, the NCL requires each member in the ensemble to be trained with gradient-descent based learning [15]. So it is difficult to apply the NCL to the CVN models which cannot be trained with a gradient based learning rule (e.g., the CVN model of [1]).

The remainder of the paper is organized as follows. In Section 2, we briefly discuss the CVN model used in the ensembles of this study, along with the classification ability of a single CVN on some Boolean problems. Two methods for creating the ensembles of single-layered CVNNs, i.e., NCL and *bagging*, are discussed in Section 3. Experimental results on a number of real-world benchmark problems are presented in Section 4. Finally, we give our concluding remarks in Section 5.

2. CVN model and its classification ability

This section briefly discusses the CVN model and its classification ability, which we presented in our earlier work [2]. The discussion includes the representation of real-valued input data to a CVN, the role of the activation function, and the classification ability of a CVN on some Boolean problems.

2.1. Input data representation

To present complex-valued information to a CVN, we encoded the real-valued data by the phases, between 0 and π, of the unity magnitude complex numbers. For example, if a real-valued number $x \in [a, b]$, where $a, b \in R$, then the corresponding complex number $z = e^{\mathbf{i}\pi(x-a)/(b-a)}$, where $\mathbf{i} = \sqrt{-1}$. Clearly, in this representation, when the real-valued variable x moves along a line from a to b, the corresponding complex variable z moves over the upper half of a unit circle on the complex plane. In order to process the Boolean data, the values TRUE and FALSE were represented by $e^{\mathbf{i}\pi}$ and $e^{\mathbf{i}0}$, respectively.

2.2. Activation function

Our motivation for designing the activation function came from the role of the activation function in a real-valued output neuron for the classification tasks. The neuron has essentially two functional parts, an aggregation part and an activation part. The aggregation part maps a multidimensional input into a one dimensional output by multiplying each of the inputs to the neuron by the connection weights and then by summing up the weighted inputs. The other part, i.e., activation function does a threshold operation on the output given by the aggregator. As for instance, consider a threshold function given by

$$y(v) = \begin{cases} \text{class } A, & \text{if } v \geqslant 0 \\ \text{class } B, & \text{otherwise,} \end{cases}$$

where $v = \mathbf{w}^{\mathrm{T}}\mathbf{x}+b$, $\mathbf{w}$ and $\mathbf{x}$ being the weight and the input vectors, and b is the bias of the neuron. Clearly, the threshold function divides its one dimensional domain into two disjoint parts; each part denotes one of the two classes.

Thus the role of an activation function, in a real-valued output neuron, is to divide the function's domain (defined by the output of the aggregator) into disjoint sets or regions for representing the corresponding classes. Hence forward, we call the output of the aggregator part as the net-input of a neuron, and the domain of an activation function as the net-input space of the neuron.

Motivated by the role of an activation function in a real-valued output neuron, we formulated a family of activation functions for CVN in [2]. The functions map complex-valued net-inputs into bounded real-values by combining the real and imaginary parts of the net-inputs. The purpose of such mapping is to divide the net-input space into different regions to represent the classes.

Fig. 1 shows one of the functions which we have used in this study of the ensembles. The function maps complex-values into bounded real values and has the form of $f_{C\to R}(u+\mathbf{i}v) = (f_R(u) - f_R(v))^2$ where $f_R(x) = 1/(1+e^{-x})$, u, v, $x \in R$, and $\mathbf{i} = \sqrt{-1}$. As can be seen from Fig. 1, the function saturates in four regions, R_1, R_2, R_3, and R_4. Among the regions, R_1 and R_3 denote one class, while R_2 and R_4 denote the other class. Note that the function is differentiable with respect to real and imaginary part of the net-input individually. Since the cost function (mean squared error) to be minimized is real-valued, this kind of differentiability is sufficient to derive a gradient-descent based learning algorithm by considering the real and imaginary parts of the weight parameters individually. See [2] and also the Section 3.1 for the details of the learning algorithm.

Liouville's theorem states that there is no complex-valued function which is bounded and differentiable in the entire complex domain except the constant. We can avoid this constraint by combining the real and imaginary part of the net-input meaningfully for the classification tasks, and this was our main objective of designing the activation functions.

2.3. Classification ability of a CVN

We studied the classification ability of a CVN with the new activation functions on several Boolean problems in [2]. Here we