Engineering Science and Technology, an International Journal 19 (2016) 197–211

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: http://www.elsevier.com/locate/jestch

Full Length Article

A novel nature inspired firefly algorithm with higher order neural network: Performance analysis

Janmenjoy Nayak *, Bighnaraj Naik, H.S. Behera

Department of Computer Science Engineering & Information Technology, Veer Surendra Sai University of Technology (VSSUT), Burla 768018, Odisha, India

ARTICLE INFO

Article history:
Received 28 March 2015
Received in revised form
1 July 2015
Accepted 7 July 2015
Available online 28 August 2015

Keywords:
Higher order neural networks (HONNs)
Pi-Sigma neural networks
Genetic algorithm (GA)
Particle swarm optimization (PSO)
Firefly algorithm (FFA)

ABSTRACT

The applications of both Feed Forward Neural network and Multilayer perceptron are very diverse and saturated. But the linear threshold unit of feed forward networks causes fast learning with limited capabilities, while due to multilayering, the back propagation of errors exhibits slow training speed in MLP. So, a higher order network can be constructed by correlating between the input variables to perform nonlinear mapping using the single layer of input units for overcoming the above drawbacks. In this paper, a Firefly based higher order neural network has been proposed for data classification for maintaining fast learning and avoids the exponential increase of processing units. A vast literature survey has been conducted to review the state of the art of the previous developed models. The performance of the proposed method has been tested with various benchmark datasets from UCI machine learning repository and compared with the performance of other established models. Experimental results imply that the proposed method is fast, steady, reliable and provides better classification accuracy than others.

1. Introduction

Data classification has been a keen interest among the data mining researchers along with other data mining tasks. Classification involves some basic steps like assigning different classes to unknown objects and identifying a suitable model to analyze them. The model of classification is based on the set of training data and the objective is to assign a unique class label distinguished from different other classes. There is a matured number of earlier classification models [1–14], those have already been developed by using some traditional neural networks like Multilayer perceptron (MLP), Back propagation neural network (BPNN), Feed forward network etc. in various diversified classification application domains.

Due to a number of drawbacks including slow training rate, late convergence, nonlinear mapping capability, etc. in some basic neural networks, higher order neural networks (HONN) became more popular in the development of real life applications. For example, with a single layer of threshold logic units, the training speed of MLPs is much slower than feed forward network and the use of the perceptron, ADALINE or Hebbian learning rules [15,16]. Also, while solving the typical complex and nonlinear problems (as the classification task of data mining is highly nonlinear), these networks have a slow converge rate and are unable to scale with problem size [17]. Moreover, MLPs require rigorous repetitive training algorithm and are prone to overfit the data [18], which causes long training time and being trapped at local minima [19]. Unlike Back Propagation Neural networks, HONN lucratively provides a proficient open-box model to map the nonlinear inputs–outputs and results in easier understanding of data mining [20]. By leaving necessity of hidden layers, HONN structures become simpler than FNNs and initialization of learning parameters (weights) will not be catastrophic. Some HONNs make use of different adaptive activation functions like Sigmoid, Sine, Cosine etc., to fit well as per the specifications of the network. Moreover, HONNs run much faster than normal feed forward network, MLP etc. [21,22]. By keeping equivalent structure as that of feed forward neural network, HONNs extend their capacity by adding input units, along with a stronger functioning of other neural units in the network. They take the advantage of nonuse of some complicated mathematical functions due to the easy transformations of input units to the hidden and other layers. The product units of HONNs increase the information capacity of the nodes in the network, whereas they possess fast learning rates due to the involvement of some higher order inequalities. Various higher order neural networks such as Sigma Pi Neural Network (SPNN) [23], Product Unit Neural Network (PUNN) [24], Higher Order Processing unit neural network (HPUNN) [25], Functional Link Artificial Neural Network (FLANN) [26] have been developed, which perform nonlinear mappings. They have been successfully used in various real life applications (Table 1) like classification, function approximation, forecasting, Time series prediction, noise control, location management, channel equalization etc. But, these higher order

* Corresponding author. Tel.: +91 9439400784, fax: 0663-2430204.
E-mail address: mailforjnayak@gmail.com (J. Nayak).
Peer review under responsibility of Karabuk University.

http://dx.doi.org/10.1016/j.jestch.2015.07.005
2215-0986/

Table 1
Different types of HONNs in various application areas.

Author(s)	Type of HONN	Application domain	Year	Reference
Mahapatra et al.	Chebyshev FLANN	Classification	2012	[27]
Naik et al.	HMBO-GDL-FLANN	Classification	2015	[28]
Mishra et al.	PSO-FLANN	Classification	2012	[29]
Dehuri et al.	IPSO-FLANN	Classification	2012	[30]
Mili and Hamdi	DE-FLANN	Classification	2013	[31]
Naik et al.	CRO-FLANN	Classification	2015	[32]
Shin and Ghosh	PSNN	Classification and function approximation	1991	[33]
Patra et al.	FLANN	Forecasting	2009	[34]
Bebarta et al.	FLANN	Forecasting	2012	[35]
Durga Ganesh Reddy and Tarun Varma	FLANN	Forecasting	2014	[36]
Ghazali et al.	RPNN	Prediction	2006	[37]
Hussain et al.	RPNN	Prediction	2008	[38]
George and Panda	FLANN	Noise control	2012	[39]
Parija et.al.	FLANN	Location management	2013	[40]
Sicuranza and Carini	FLANN	Noise control	2012	[41]
Ali and Haweel	FLANN	Channel equalization	2013	[42]

networks require some higher order terms, when the order of network becomes exceptionally high and affects the complexity of the networks. Shin and Ghosh developed a HONN, called Pi-Sigma neural network (PSNN), which avoids the exponential increase of no. of weight vectors along with the processing units. The PSNN uses the product of sum of input components having the linear summation of a single hidden layer and the product of processing units at output layer, instead of sum of product of inputs as other networks.

Over the decades several versions of PSNN have been developed for various application domains by different researchers. A.J. Hussain and P. Liatsis [43] have introduced a recurrent pi-sigma neural network (RPSN), used as predictor structure in Differential Pulse Code Modulation systems that utilizes both the temporal dynamics of the image formation process and the multi-linear interactions between the pixels for 1D/2D predictive image coding. A memory based Sigma-Pi-Sigma neural network for excellent learning convergence along with reducing the memory size and overcoming the possible extensive memory requirement problem has been suggested by Chien-Kuo Li [44]. R. Ghazali et al. [37] have proposed a Ridge polynomial network (RPN) for financial time series prediction by adding different degrees of Pi-Sigma neural networks. It was able to find an appropriate input–output mapping of various chaotic financial time series data with a good performance in learning speed and generalization capability. A sigma-pi network trained with an online learning algorithm for solving the frame of reference transformation problem has been presented by Cornelius Weber and Stefan Wermter [45]. An online gradient algorithm for Pi-Sigma neural networks with stochastic inputs with improved computational efficiency have been proposed by X. Kang et al. [46]. A switch reluctance motor based on Pi-Sigma neural network has been developed by Jie Xiu and Chang-Liang Xia [47] and the tested results demonstrate high accuracy, strong ability of generalization, and fast computational speed of the model. Yong Nie and Wei Deng [48] realized that hybrid genetic algorithm can search out the global optimum which is faster than genetic algorithm and their proposed hybrid genetic algorithm trained Pi-Sigma network was used to resolve the function optimization problem. Ge Song et al. [49] have proposed a new visual cryptography scheme for general access structures using pi-sigma neural networks to infuse a new activity in visual cryptography researching. A special class of HONN based Pi-sigma networks, trained by distributed evolutionary algorithms with the global optimization methods, has been studied by M.G. Epitropakis et al. [50]. A novel hybrid higher order neural classifier for handling classification problems has been proposed by M. Fallahnezhad et al. [20] by considering a number of benchmark datasets to get the improved accuracy results. Xin Yu and Qingfeng Chen [51] have trained the Ridge Polynomial neural network with gradient algorithm having synchronous update rule and penalty term. They claim for the Monotonicity and strong convergence of their developed method. A Monotonicity theorem and two convergence theorems of the asynchronous gradient method for training the ridge polynomial neural network have been proposed by Xin Yu et al. [52] to perform effective training. Nayak et al. [53] have proposed a standard back propagation Gradient descent learning based Pi-sigma neural network for data classification with global searching capabilities.

In this study, the performance of a class of higher order neural network called Pi-Sigma neural network has been tested by using a recently developed nature inspired metaheuristic algorithm such as firefly algorithm (FFA). To study the performance of the FFA, we have considered some standard typical benchmark datasets and the performance has been compared with other optimization algorithms like GA, PSO and Hybrid GA-PSO on the same datasets.

The remainder of this paper is organized as follows. Section 2 reviews some basic preliminaries like Pi-Sigma network, Firefly Algorithm, Particle Swarm Optimization and Genetic Algorithm. In Section 3, the proposed FF based PSNN has been presented. Experimental setup and Results Analysis have been presented in Sections 4 and 5 respectively. Section 6 is devoted to Cross Validation and Section 7 concludes the work with some future directions.

2. Preliminaries

2.1. Pi-sigma neural networks (PSNNs)

The output of the network is computed by the product of sum operation of the input units at the output layer. The output of the network is computed by the product of sum operation of the input units at the output layer. Fewer weight vectors and processing units are capable of quick learning which makes them more accurate and tractable than the other networks. The weights connected from the input layer to the hidden layer are tailored during the training and the weights connecting the hidden layer to the output layer are fixed to unity. Due to this reason the complexity of the hidden layer can be dramatically reduced by the number of tunable weights, for which the model can be easily implementable and accelerated [54,55] (Fig. 1).

Let the input $x = (x_0, x_1 \ldots, x_j \ldots x_n)^T$ be the $(n+1)$ dimensional input vectors where additional B_j is the bias unit and x_j denotes the j^{th} component of X. The $(n+1)k$ dimensional weight vectors such that $w_{ij} = (w_{ij0}, w_{ij1}, w_{ij2} \ldots w_{ijn})^T$, $i = 1,2 \ldots k$ are summed at a layer of k summing units, where k is the corresponding order of the network. The output at the hidden layer h_j can be computed by Eq. (1).

$$h_j = B_j + \sum w_{ji} x_i \tag{1}$$

where w_{ij} represents the weight from the input to summing unit. As the weight in the hidden layer to the output layer is fixed to 1, the output O can be computed by Eq. (2).

$$O = f\left(\prod_{j=1}^{k} h_j\right) \tag{2}$$

where $f(.)$ is a suitable activation function. The order of the PSNN can be computed by the exact number of processing neurons in the hidden layer. The structure of the network may be regularly