Expert Systems With Applications 56 (2016) 80–88

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Robust learning algorithm for multiplicative neuron model artificial neural networks

Eren Bas[a,*], Vedide Rezan Uslu[b], Erol Egrioglu[a]

[a] Giresun University, Faculty of Arts and Science, Department of Statistics, Gure Campus, Gure Area Giresun 28100, Turkey
[b] University of Ondokuz Mayis, Faculty of Arts and Science, Department of Statistics, Kurupelit Campus, Kurupelit Samsun 55139, Turkey

ARTICLE INFO

Article history:
Received 4 April 2015
Revised 27 February 2016
Accepted 28 February 2016
Available online 8 March 2016

Keywords:
Artificial neural networks
Multiplicative neuron model
Particle swarm optimization
Robust learning algorithm

ABSTRACT

The two most commonly used types of artificial neural networks (ANNs) are the multilayer feed-forward and multiplicative neuron model ANNs. In the literature, although there is a robust learning algorithm for the former, there is no such algorithm for the latter. Because of its multiplicative structure, the performance of multiplicative neuron model ANNs is affected negatively when the dataset has outliers. On this issue, a robust learning algorithm for the multiplicative neuron model ANNs is proposed that uses Huber's loss function as fitness function. The training of the multiplicative neuron model is performed using particle swarm optimization. One principle advantage of this algorithm is that the parameter of the scale estimator, which is an important factor affecting the value of Huber's loss function, is also estimated with the proposed algorithm. To evaluate the performance of the proposed method, it is applied to two well-known real world time series datasets, and also a simulation study is performed. The algorithm has superior performance both when it is applied to real world time series datasets and the simulation study when compared with other ANNs reported in the literature. Another of its advantages is that, for datasets with outliers, the results are very close to the results obtained from the original datasets. In other words, we demonstrate that the algorithm is unaffected by outliers and has a robust structure.

1. Introduction

Making an inference about the future, i.e., forecasting, plays an active role in a country's economy and politics. In this respect, many models have been proposed in the literature to deal with forecasting issues. Classical forecasting models may be insufficient to solve queries of datasets because they require several assumptions. In recent years, artificial neural networks (ANNs) that do not utilize assumptions valid for classical forecasting models have been used extensively and successfully as an alternative to the classical time-series approaches.

ANNs are generally defined as mathematical algorithms inspired by biological neural networks that can learn from the samples and generalize them. It can be said that the most important feature of ANNs is their ability to learn from a source of information (data). The learning process of an ANN is the process of finding the best values of weights that connects neurons and this process is called "training". Finding the best weights in an ANN can be considered as an optimization problem. Some of the most used learning algorithms are the Levenberg–Marquardt (LM) and Back Propagation (BP) learning algorithms. LM and BP algorithms involve methods based on derivatives. Also, these methods have been used in the training of ANNs before artificial intelligence optimization algorithms. Therefore, approaches based on derivatives such as LM and BP was used in many studies in the literature.

There are, however, other methods used in the training of an ANN including heuristic algorithms such as genetic algorithms, particle swarm optimization (PSO), simulated annealing, and taboo search algorithms. The first ANN model was proposed by McCulloch and Pitts (1943). The studies on ANN began with a single-layer neural network, which consisted of only input and output layers and an output function for the single-layer neural networks which was linear.

However, single-layer neural networks were unable to solve non-linear problems. In the interim, to solve non-linear problems, the multilayer perceptron (MLP) model was developed by Rumelhart, Hinton, and Williams (1986). A MLP model consists of an input, one or more hidden layers, and an output layer. The main objective of the MLP model is to reduce the difference between the output produced by the network and with the expected output of the network (error). More recently, several ANN

* Corresponding author. Tel.: +90 454 3101400; fax: +90 454 3101477.
E-mail addresses: eren.bas@giresun.edu.tr, eren6061@hotmail.com.tr (E. Bas), rezzanu@omu.edu.tr (V.R. Uslu), erol.egrioglu@giresun.edu.tr (E. Egrioglu).

http://dx.doi.org/10.1016/j.eswa.2016.02.051

models were proposed by Basu and Ho (1999), Labib (1999), Plate (2000), and Zhang et al. (2000). In addition to these types of ANN, there are several others in the literature such as the generalized mean neuron (GMN) model proposed by Yadav, Kumar, Kalra, and John (2006) and geometric-mean neuron (G-MN) model proposed by Shibblee, Chandra, and Kalra (2010). However, the aggregation functions based on the average of all these models are affected quite negatively when the dataset has outliers.

In this case, the forecasts of the models move away from the general characteristics of the data and tend towards outliers. Hence, the model will give lower or higher estimates than necessary depending on whether the outlier is high or low. In addition, it was shown that time series containing outliers is incompatible with the general characteristics of data, and negatively affects the forecasting performance of the MLP in the studies of Hill, Marquez, O'Connor, and Remus (1994) and Zhang, Eddy Patuwo, and Hu (1998). However, Chuang, Hsiao, and Su (2000) showed that even classical ANN approaches have problems when the dataset has outliers.

Subsequently, several studies reported in the ANN literature have been able to reduce the effects of outliers or achieve good results even in the presence of outliers using robust methods. The statistics generating forecasts with good properties despite deviations from the assumptions are called robust statistics.

The history of robust regression estimators is founded on the least absolute deviations or L_1 estimation method proposed by Boscovich (1757). Studies on robust estimators have resurged again with the development of computer programs. Huber (1964) and Andrews (1974) studied different methods; in particular, Huber's M-type estimators form the basis for robust regression analysis. In the ANN literature, there are several robust learning algorithms that can be used even with the presence of outliers in the dataset. In general, these studies have used the M estimation method as fitness function. The basic idea of the M estimators in ANN methods is based on replacing the lost function and reducing the effect of outliers and square error term. However, there are some problems in the use of these robust approaches proposed in the ANN literature. The first of these is the choice of initial weight, and the second is the choice of the parameter of the scale estimator (c) in M estimators. The parameter of the scale estimator is very important in reducing the effects of outliers as they may not be reduced sufficiently if this parameter is large; if this parameter is taken to be small, values that are not actually outliers may be detected as such.

In the literature, the scale estimator is calculated based on robust statistics such as the median of errors or median of the absolute deviation. While some of robust methods suggested in ANN literature focus on learning process, others focus on architecture. Chen and Jain (1994) and Hsiao, Chuang, and Jeng (2012) used the M-estimator as a fitness value in the learning algorithm. In addition, Lee, Chung, Tsai, and Chang (1999) proposed a robust learning algorithm for radial basis neural networks. Espinoza, Ordieres, Mertinez, and Gonzalez (2005) proposed a robust feedback learning algorithm to overcome the problems inherent in robust methods that have been used in the ANN literature. El-Melegy, Essai, and Ali (2009) and Rusiecki (2012) proposed a robust learning algorithm based on the median of squares. Aladag, Yolcu, and Egrioglu (2014) have proposed a robust neural network based on the median neuron model although in this study the learning algorithm is not robust whereas the architecture is. Chuang et al. (2000) suggested a robust feedback learning algorithm that gives effective results both in the absence and presence of outliers and also this proposed method was proposed to overcome problems in classic robust learning algorithms. Allende, Moraga, and Salas (2002) proposed a robust estimator for the ANN learning process used in time series applications. Neubauer (1995) used a robust learning algorithm for Li (2009) proposed a learning algorithm for the training of a radial basis neural network with PSO. Juanyin, Wei, and Pan (2004) proposed a robust learning algorithm for the ensemble of ANN. Hans and Udluft (2010) proposed that the ensemble of ANNs should be used to make the learning process more robust.

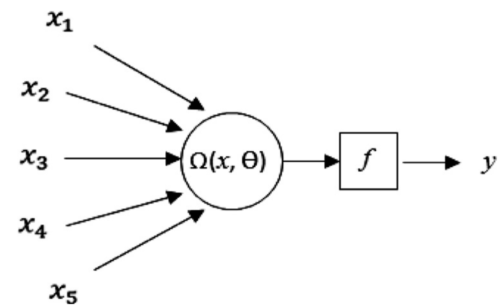

Fig. 1. Multiplicative neuron model with five inputs.

Aside from these studies that looked at overcoming the problems of learning algorithms, there are some robust learning algorithms in the literature proposed by Connor, Martin, and Atlas (1994), Liano (1996), McDowall and Ham (1997), Mili et al. (1996), Sanchez (1995), Umasuthan and Wallace (1996), Wang, Wu, and Principe (1996), and Huang, Zhang, and Huang (1998). There are also some robust learning algorithms proposed for multi-layer feed-forward neural network (ML-FF-ANN) as previously mentioned. There is no robust learning algorithm in the literature for multiplicative neuron model (MNM) ANN.

In this paper, a robust learning algorithm is proposed for MNM-ANN. The method involves the training of MNM-ANN performed by PSO using the Huber loss function as fitness function. Hence, it is concluded that the forecasting performance was enhanced using the robust learning algorithm and gives better forecasting results when compared with other methods both with and without outliers. In addition, the parameter of the scale estimator which governs the Huber loss function is also estimated with the proposed method.

The remainder of this paper is organized as follows: The fundamentals of the multiplicative neuron model (MNM) are given in Section 2. PSO is briefly summarized in Section 3. The proposed method is introduced in Section 4. Applications are presented and studied in Section 5. First, results obtained from the application of the proposed method are examined by considering different real life time series and then the performance of the proposed method were statistically evaluated with a simulation study. Section 6 presents discussions and conclusions.

2. Multiplicative neuron model

The single multiplicative ANN, which is a special case of the neuron model, was first presented by Yadav, Kalra, and John (2007). This model had only one neuron; the general structure of the model is illustrated in Fig. 1. In this model, x_i $(i = 1, \ldots, 5)$ is the input pattern and when output values are computed a multiplication function is used instead of a sum function.

The operator $\Omega(x, \theta)$ is a multiplicative operation which involves a multiplication of weighted inputs. The activation function and target of the model are represented by f and y, respectively. For training the MNM, there are several learning algorithms. The BP and cooperative PSO learning algorithms were used in processes for the MNMs by Yadav et al. (2007) and by Zhao and Yang (2009), respectively.