Neurocomputing 207 (2016) 676–683

Contents lists available at ScienceDirect

Neurocomputing

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

Mutual information based weight initialization method for sigmoidal feedforward neural networks

Junfei Qiao [a,b,*], Sanyi Li [a,b], Wenjing Li [a,b]

[a] College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, China
[b] Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China

ARTICLE INFO

Article history:
Received 27 January 2016
Received in revised form
7 April 2016
Accepted 24 May 2016
Communicated by Ma Lifeng Ma
Available online 4 June 2016

Keywords:
Sigmoidal feedforward neural network
Weight initialization
Mutual information

ABSTRACT

When a sigmoidal feedforward neural network (SFNN) is trained by the gradient-based algorithms, the quality of the overall learning process strongly depends on the initial weights. To improve the algorithm stability and avoid local minima, a Mutual Information based weight initialization (MIWI) method is proposed for SFNN. The useful information contained in input variables is measured with the mutual information (MI) between input variables and output variables. The initial distribution of weights is consistent with the information distribution in the input variables. The lower and upper bounds of the weights range are calculated to ensure the neurons inputs are within the active region of sigmoid function. The MIWI method makes the initial weights close to the global optimal point with a higher probability and avoids premature saturation. The efficiency of the MIWI method is evaluated based on several benchmark problems. The experimental results show that the stability and accuracy of the proposed method are better than some other weight initialization methods.

1. Introduction

Neural network can be considered as a nonlinear system and the training process can be seen as a nonlinear optimization process [1,2]. Due to the fact that the optimal weights are difficult to be found with analytical methods, iterative local or global optimization methods are necessary [3]. The gradient-based (GB) training algorithms are widely used due to its effectiveness [3]. However, GB methods have slow convergence rate and are often hampered by the occurrence of local minima [4]. It has been demonstrated that the performance (convergence rate and training accuracy) of sigmoidal feedforward neural network (SFNN) trained with GB algorithms depends on several factors, including training algorithm, initial conditions, training data and network structure [5–8]. Use of appropriate weight initialization can shorten the training time and avoid the local minima caused by random initial weights [9–22].

Many methods have been developed to choose the suitable initial weights of SFNN. Generally speaking, they can be classified into two categories, namely, least squares (LS) methods and interval analysis (IA) methods. The LS methods calculate accurate initial weights to diminish the initial error, which has been employed in many literatures. Yam [9] proposed a weight initialization algorithm based on a linear algebraic, and the optimal initial weights of each layer are evaluated by LS method. Erdogmus [10] proposed a backpropagation of the desired response algorithm that approximates the nonlinear least squares problem with linear least squares. Erdogmus [3] improved the optimization accuracy by further considering the local scaling effects of the nonlinearities. Liu [11] used the partial LS algorithm to set the initial weights and the optimal number of hidden neurons simultaneously. Timotheou [1] approximated the nonlinear equations of the network to obtain linear equations with nonnegativity constraints, and then developed a projected gradient algorithm to solve the formulated linear nonnegative LS problem. These algorithms can diminish the initial error effectively, but cannot avoid local minima and the performances of those methods are unstable.

The IA methods determine optimal range for the initial weights and biases to ensure the hidden neurons be active or avoid premature saturation. The optimal range is investigated from different aspects. Drago [20] obtained the maximum magnitude of the weights based on statistical analysis to improve the convergence speed. Thimm [15] designed many experiments to determine the optimal range for the initial weights and biases of high-order networks. Yam [18] determined the initial weights based on the Cauchy's inequality and proposed a linear algebraic method to guarantee the outputs of the neurons within the active region. In subsequent research, Yam [19] proposed a method based on multidimensional geometry, which ensured the activation function be fully utilized.

* Corresponding author at: College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, China.
E-mail address: junfeiq@bjut.edu.cn (J. Qiao).

http://dx.doi.org/10.1016/j.neucom.2016.05.054
0925-2312/

Yang [22] proposed a weight initialization method based on the theory that the range of the initial values should be greater than the adjustment quantities. Adam [21] proposed a linear interval tolerance approach to guarantee the input of each hidden neuron be in the active region of sigmoid function. Talaska [17] proposed an initialization mechanism based on a Convex Combination method that is easy to be realized in transistor. Although the convergence rate with these methods can be accelerated, the performance is unstable and cannot avoid local minima.

In order to solve the stability problem, we propose a Mutual Information based weight initialization (MIWI) method in this paper, which takes advantage of mutual information (MI) in evaluation of information between variables. MI has been demonstrated to be effective in neuron network to measure the interrelation of two neurons. For example, Qiao [23] added or removed hidden neurons of RBF neural network based on the MI between hidden neurons and output neurons. Zhang [24] proposed an adaptive merging and splitting algorithm for feedforward neural networks based on MI. Peng [25] proposed the minimal-redundancy-maximal-relevance criterion based on MI to select optimal hidden neurons. Puma-Villanueva [44] proposed a constructive algorithm for feedforward neural network based on MI. Chen [26] proposed a clustering algorithm based on partial mutual information (PMI) to implement clustering neurons. Besides some other researchers [27–29] also applied PMI to selecting input variables of artificial neural network.

In this paper, the MIWI algorithm adopts the MI between input variables and output variables to measure the useful information contained in input variables (the neurons with high MI contain more useful information). The MIWI is mainly based on three theorems that have been verified:

(1) If the initial weights are close to the global optimum, any descent algorithm can train the weights toward the optimum reliably [3];
(2) The network with optimal weights can fully reflect the relationship between the input variables and output variables [17];
(3) Restrain the range of initial weights reasonably can shorten the training time dramatically [16,21].

The MIWI algorithm has several advantages as follows: Firstly, MI is calculated between each input variable and output variable. Input variable with high MI means the information contained is more important. Since the sigmoid function is a monotone increasing function, the initial weights between input neurons and hidden neurons are positively related to the mutual information of the input neuron they connected. Thus the initial distribution of the weights is consistent with the information distribution in the input variables. To ensure the diversity of hidden neurons, the biases of hidden neurons are distributed in the value space uniformly and randomly. Thus the initial weights have higher probability to get close to the optimal point and the performance can be more stable than by random initialization.

Secondly, based on the distribution of the initial weights, the lower and the upper bounds of the initial weights are calculated to guarantee all the hidden neurons are active in the initial phase. Then the convergence rate can be guaranteed and the premature saturation can be avoided.

This paper is arranged as follows. The basic conceptions are introduced in Section 2. The MIWI algorithm is described particularly in Section 3. The experimental results and the performance comparison between MIWI and other weight initialization methods are presented in Section 4. A discussion of the merits of the proposed MIWI algorithm is given in Section 5. The conclusion is presented in Section 6. For convenience of discussion, the acronyms used in this paper are listed in Table 1.

Table 1
Lists of acronyms.

Acronym	Description
SFNN	Sigmoidal feedforward neural network
BP	Backpropagation
GB	Gradient-based
MIWI	Mutual Information based weight initialization
MI	Mutual information
LS	Least squares
IA	Interval analysis
PMI	Partial mutual information
MLPs	Multilayer perceptions
SCBP	Split-complex backpropagation algorithm
ETP	Effluent total phosphorus
T	Temperature
ORP	Oxidation reduction potential
DO	Dissolved oxygen concentration
TSS	Total soluble solid
pH	Potential of hydrogen
ITP	Influent total phosphorus

2. Basic conceptions

In this section, we briefly introduce SFNN and MI.

2.1. Sigmoidal feedforward neural networks (SFNN)

The SFNN is a kind of multilayer perceptions (MLPs) that applies sigmoidal activation functions in hidden neurons. Due to the MLPs with one hidden layer can approximate any continuous function [30,31], we mainly research the one-hidden-layer SFNN. The structure of a one-hidden-layer SFNN is shown in Fig. 1.

In this paper, let n_0,n_1, and$n_2$denote the number of neurons in the input layer, the hidden layer and the output layer respectively. The input and output of the three layers are $[z^{(0)}, y^{(0)}]$, $[z^{(1)}, y^{(1)}]$ and $[z^{(2)}, y^{(2)}]$, respectively. Assume the input vector is $X = [x_1, x_2, \ldots, x_{n_0}]$, then the output of the ith input neuron can be expressed as

$$y_i^{(0)} = x_i, (i = 1, 2, \ldots, n_0) \tag{1}$$

In hidden layer, each neuron is connected to all the input neurons. The input and output of the jth hidden neuron are

$$z_j^{(1)} = \sum_{i=1}^{n_0} w_{ij}^{(1)} y_i^{(0)} + b_j^{(1)}, (j = 1, 2, \ldots, n_1) \tag{2}$$

$$y_j^{(1)} = f^{(1)}\left(z_j^{(1)}\right) = \left(1 + \exp\left(-z_j^{(1)}\right)\right)^{-1} \tag{3}$$

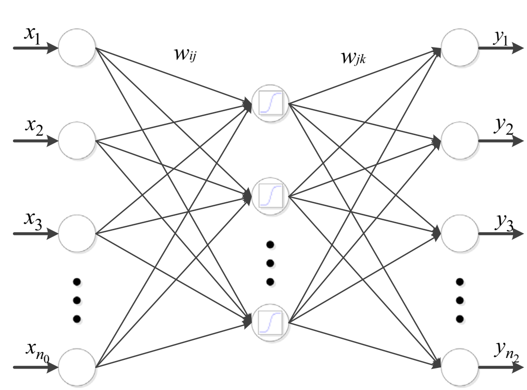

Fig. 1. The one-hidden-layer SFNN structure.