Neurocomputing 143 (2014) 182–196

Contents lists available at ScienceDirect

Neurocomputing

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

Bi-modal derivative activation function for sigmoidal feedforward networks

Sartaj Singh Sodhi, Pravin Chandra *

University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University, Dwarka Sector – 16C, Delhi 10078, India

ARTICLE INFO

Article history:
Received 29 October 2013
Received in revised form
14 March 2014
Accepted 5 June 2014
Communicated by M.T. Manry
Available online 17 June 2014

Keywords:
Sigmoidal function class
Bi-modal activation function
Activation function
Sigmoidal feed-forward artificial neural network

ABSTRACT

A new class of activation functions is proposed as the sum of shifted log-sigmoid activation functions. This has the effect of making the derivative of the activation function with respect to the net inputs, be bi-modal. That is, the derivative of the activation functions has two maxima of equal values for nonzero values of the parameter, that parametrises the proposed class of activation functions. On a set of ten function approximation tasks, the usage of the proposed activation function demonstrates that there exists network(s), using the proposed activation, and are able to achieve lower generalisation error, in equal epochs of training using the resilient backpropagation algorithm. On a set of four benchmark problems taken from UCI machine learning repository, for which the networks are trained using the resilient backpropagation algorithm, the scaled conjugate algorithm, the Levenberg–Marquardt algorithm and the quasi-Newton BFGS algorithm, we observe that the usage of the proposed algorithms leads to better generalisation results, similar to the results for the ten function approximation tasks wherein the networks were trained using the resilient backpropagation algorithm.

1. Introduction

Artificial Neural Networks (ANNs) have been demonstrated to be an efficient technique for the solution of complex tasks or learning problems. The set of all learning problems range over and can be broadly be subdivided into four classes of common problems [12]:

1. Classification
2. Regression
3. Density Estimation
4. Clustering/Vector Quantisation.

To a large extent, the success of the feedforward artificial neural networks (FFANN) with sigmoidal hidden nodes, in solving complex learning tasks, may be attributed to the possession of the Universal Approximation Property (UAP) by these networks. These results imply that for an appropriate non-linearity at the hidden layer of a three layer network,[1] the network with sufficient number of hidden nodes can approximate any continuous function arbitrarily well [6,14,16,21,24,32]. A single hidden layer FFANN is represented as (Fig. 1)

1. The inputs are to be taken in a finite interval [32], usually [0,1] or [−1,1].
2. Fig. 1 shows the schematic of a single hidden layer network with one output node.[2]
3. The input nodes transfer the input(s) without any processing.
4. The hidden layer net input (for the ith node) is given by:

$$n_i = \sum_{j=1}^{\mathcal{I}} w_{ij} x_j + \theta_i; \quad i \in \{1, 2, \ldots, \mathcal{H}\} \tag{1}$$

where $\mathcal{I}$ is the number of inputs, and $\mathcal{H}$ is the number of hidden nodes.
5. The activation function $\phi(\cdot)$ is a map $\phi : \mathcal{R} \to \mathcal{R}$, where $\mathcal{R}$ is the set of all real numbers. The activation function is used as the

* Corresponding author.
E-mail addresses: sartaj@ipu.ac.in, sartajsodhi@yahoo.com (S. Singh Sodhi), chandra.pravin@gmail.com, pchandra@ipu.ac.in (P. Chandra).

[1] The first layer is called the input layer, the last layer is known as the output layer and the layer in-between the input(s) and the output(s) is known as the hidden layer. Thus, a three layer network may also be known as a single hidden layer network.

[2] Though the network may have one or more layers, in between the input and the output layer, we consider the single hidden layer network as the minimal network to have the UAP.

http://dx.doi.org/10.1016/j.neucom.2014.06.007
0925-2312/

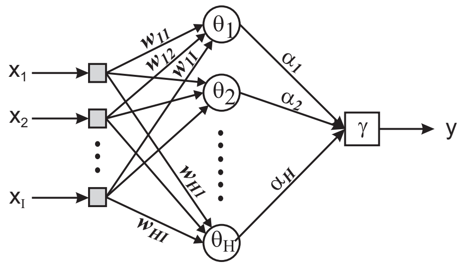

Fig. 1. The schematic diagram of a single hidden layer network.

hidden layer node non-linearity (to transform the net input to the output of the node). The properties generally required to be fulfilled by these activation functions are:

(a) The function must be bounded [6,14,16,21,24].
(b) The function should be continuous [6,14,16]. This together with the condition of the differentiability is a condition that is also required by the training algorithms based on gradient, hessian and/or conjugate direction calculation (s) or where the derivative of the activation function is required for the calculation of the weight updates during training. Though for the existence of the UAP, this requirement is not necessary [21,24].
(c) The function should be a sigmoid [6,14,16,21,24]. Or, the limits for $\pm\infty$ for the function satisfy the following [9]:

$$\lim_{x\to-\infty}\phi(x)=\alpha \tag{2}$$

$$\lim_{x\to\infty}\phi(x)=\beta \tag{3}$$

with $\alpha<\beta$.
(d) The output from the kth output node being:

$$y_k=\sum_{i=1}^{\mathcal{H}}\alpha_{ki}\phi_i(n_i)+\gamma_k;\quad k\in\{1,2,\ldots,\mathcal{O}\} \tag{4}$$

where $\mathcal{O}$ is the number of output nodes, and ϕ_i is the non-linear map also known as the activation function or squashing function at the ith hidden layer node.
(e) The condition of monotonicity of the activation function is not a mandatory requirement for the existence of the UAP [14,24].

The cardinality of the class of activation function(s), satisfying these requirements, is infinite [7,15].

6. The number of nodes present is not specified by these UAP results, the general assertion being that – if sufficient number of hidden nodes, satisfying the above properties for the activation function exist, the network may approximate any continuous function arbitrarily well. The currently best known bounds for the achievable error for a network with N nodes is the bound given in [3] as $\mathcal{O}(N^{-1/2})$ or the result reported in [29] as $\mathcal{O}\left(N^{-1/4}\right)$ (see also, [4,28,36,39,41]).[3]

Thus, the fundamental choices to be made while choosing the architecture of a single hidden layer network are[4]:

1. The choice of number of nodes in the hidden layer. This problem is usually solved by running a set of experiments where the number of nodes in the hidden layer is varied for a few instances of the network, and if a network of a particular size gives a satisfactory approximation, the size is fixed. That is, the network size (or the size of the hidden layer) is fixed by exploratory experiments.
2. The choice of the activation function. This choice is usually made on the basis of the preference of the researcher. Though some guidelines for the preference of anti-symmetric (or odd) functions have been given [20,26]. The commonly utilised activation functions are:
   (a) The logistic or the log-sigmoid function, defined as

$$\sigma_l(x)=\frac{1}{1+e^{-x}} \tag{5}$$

The derivative of this function is

$$\frac{d\sigma_l(x)}{dx}=\sigma_l'(x)=\sigma_l(x)(1-\sigma_l(x)) \tag{6}$$

   (b) The hyperbolic tangent function, defined as

$$\sigma_h(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}=2\sigma_l(2x)-1 \tag{7}$$

The derivative of this function is

$$\frac{d\sigma_h(x)}{dx}=\sigma_h'(x)=(1-\sigma_h^2(x)) \tag{8}$$

For a summary of universal approximation results for FFANNs with sigmoidal hidden nodes and further references see [10,32,35]. For a survey of activation functions see [15].
These usually utilised activation functions derivatives are uni-modal[5] in nature (see Fig. 2). The uni-modality of the derivative of the activation function is not a requirement for the possession of the UAP by the FFANN. In this paper we explore the feasibility of utilising a bi-modal derivative activation function that is sigmoidal, monotonically non-decreasing, continuous and differentiable.

The paper is organised as follows: in Section 2 we describe the properties of the activation function used. Section 3 describes the experimental design. Section 4 presents the results while conclusions are presented in Section 5.

2. Bimodal activation function

We define a sigmoidal function, in this paper, as:

Definition 1. A sigmoidal function $\sigma(\cdot)$ is a map $\sigma:\mathcal{R}\longrightarrow\mathcal{R}$, where $\mathcal{R}$ is the set of all real numbers, having the following limits for the argument to the function (x) tending to $\pm\infty$:

$$\lim_{x\to\infty}\sigma(x)=\beta \tag{9}$$

$$\lim_{x\to-\infty}\sigma(x)=\alpha;\quad\alpha<\beta \tag{10}$$

[3] In practice, it is observed that these bounds are very loose; that is, convergence of training to achieve the desired (error) goal is obtained with substantially lesser number of nodes as compared to the number of nodes predicted by these bounds. In practice, the number of nodes, in the hidden layer (s), is fixed by exploratory experiments wherein the number of nodes in the hidden layer is varied. The smallest number of nodes, that gives a "tolerable" error is taken as the size of the hidden layer(s).

[4] It may be kept in view that the choice of the initial configuration of weights is also a question to be answered before the training of any network is begun, for the discussion and in context to this paper, it is relevant only in the fact that there must be an initial set of weights as per the requirement of the training algorithm.

[5] The term uni-modal here is used with the meaning that a uni-modal function has only one local maxima. Thus, a bi-modal function has two local maxima.