Taylor & Francis
Taylor & Francis Group

IETE Journal of Research

ISSN: 0377-2063 (Print) 0974-780X (Online) Journal homepage: https://www.tandfonline.com/loi/tijr20

New Algebraic Activation Function for Multi-Layered Feed Forward Neural Networks

K. V. Naresh Babu & Damodar Reddy Edla

To cite this article: K. V. Naresh Babu & Damodar Reddy Edla (2017) New Algebraic Activation Function for Multi-Layered Feed Forward Neural Networks, IETE Journal of Research, 63:1, 71-79, DOI: 10.1080/03772063.2016.1240633

To link to this article: https://doi.org/10.1080/03772063.2016.1240633

Published online: 28 Oct 2016.

IETE JOURNAL OF RESEARCH, 2017
VOL. 63, NO. 1, 71–79
http://dx.doi.org/10.1080/03772063.2016.1240633

New Algebraic Activation Function for Multi-Layered Feed Forward Neural Networks

K. V. Naresh Babu and Damodar Reddy Edla

Department of Computer Science and Engineering, National Institute of Technology Goa, Ponda, India

ABSTRACT
Number of attempts have been made in the past to improve the generalization performance of artificial neural networks (ANN). Internal adjustment in ANN deals with tuning of various parameters like learning rate, activation function, etc. Majority of activation functions that exist in the literature are transcendental in nature. In this paper, a novel parametric algebraic activation (PAA) function has been proposed. The function PAA is a generalized function of which Elliott activation function is a special case. PAA is a family of S-shaped curves and satisfies all the important properties of activation functions. This activation is employed on resilient propagation algorithm (RPROP) learning algorithm. Comparative performance evaluation with the widely known activation functions has been carried out on various benchmark datasets taken from University of California Irvine (UCI) machine learning repository. Comparative performance evaluation in terms of number of epochs and testing error of the proposed PAA has been made with the standard activation functions. It has been observed that decrease in number of epochs and testing error for the proposed PAA in RPROP is highly statistically significant for most of the UCI datasets when compared with standard activation functions. Thus incorporating PAA in RPROP can make it more powerful for classification.

KEYWORDS
Algebraic activation function; Neural network; Resilient Propagation

1. INTRODUCTION

In biologically plausible neural networks, the activation function represents the rate of action potential firing in the cell. Empirically, this function is binary – either the neurons fires or does not. One of the important components of a neuron in artificial neural networks (ANN) is the activation function which alters the output of a neuron that in turn affects the performance of the neural network [1–3]. Sigmoid [4], Tan hyperbolic [5], and step functions [5] are widely used as activation functions in ANN. Combination of various activation functions such as polynomial, periodic, and Gaussian functions has also been proposed in multi-layered feedforward neural networks [6]. The sigmoid function is known as sigmoidal curve [7] or logistic function and which produces an "S" shape curve. It has been used as a threshold function in neural networks [8]. It has also been used as an approximation of Gaussian probability distribution, logistic regression [9], or membership function in fuzzy theory [10]. Extensions and variations of sigmoid as activation function have been attempted by various authors [11,12]. Sigmoidal function has also been used to approximate continuous functions. The approximation properties of the right sigmoidal and hyperbolic tangent are discussed [13]. It has been proved that feedforward networks with one layer of sigmoidal nonlinearities achieve an integrated squared error which is inversely proportional to number of nodes. The nonlinear parameters associated with the sigmoidal nodes, as well as the parameters of linear combination, are adjusted in the approximation [14]. These approximations have also been generalized and expressed in terms of the Fourier transform, by superpositions of a fixed sigmoidal function [15]. A class of quasi-interpolation operators with logarithmic sigmoidal function is constructed for approximating continuous functions defined on the total real axis [16]. Single hidden layer feedforward neural networks with ramp sigmoidal activation functions are constructed to approximate two-variable functions defined on compact interval [17]. Accurate and parsimonious approximations for indicator functions of d-dimensional balls are explained using level sets associated with the thresholding of a linear combination of ramp sigmoid activation functions [18]. Series of sigmoidal functions have been used for approximating absolutely continuous functions and the relation with neural networks approximation is discussed [19]. Max-product neural network operators in a Kantorovich-type version have been developed, which is suitable in order to study the case of Lp-approximation for not necessarily continuous data. Moreover, also the cases of the pointwise and uniform approximation of continuous functions have also been discussed [20]. Superposition of a sigmoidal function has been used for approximation error to a continuous function defined on the whole real line. It has been proved that the approximation order by $3n$ superposition of a sigmoidal function is $O(1/n)$. [21]. Sigmoid function has also been used for Hausdorff approximations, fuzzy set theory, cumulative distribution functions, etc. [22]. Sigmoidal function has also been used to solve integral equation. Volterra integral equations of the second kind have also been solved using sigmoidal functions approximation [23]. A collocation method has been discussed using sigmoidal functions for solving nonlinear Volterra integro-differential equations of neutral type [24]. The interpolation of multivariate data by operators of the neural network type is discussed. The neural network type operators have also been used to approximate continuous functions defined on a box-domain of R^d [25].

Elliot activation function, an algebraic activation function, has been used for Gaussian shape function approximation [26,27]. Sine activation function [28] has also been proved to be faster than sigmoid activation function. Algebraic activation functions other than Elliot functions are discussed for multi-layered feedforward neural networks [29]. It is also been shown that inverse tan activation outperforms logarithmic activation [30].

Optimization of activation functions for ANN is discussed [31]. Kernel extreme learning machine algorithm based on self-adaptive artificial bee colony optimization algorithm uses combination of Gaussian function as an activation function. By using bihyperbolic functions in place of sigmoid function, it has been observed that the backpropagation algorithm performs better than the traditional sigmoid functions [32]. Exponential decay activation function has been used in multi-layer perceptron for airwaves estimation [33]. Hermite activation function has been used in modern automatic speech recognition system to adjust its parameters [34].

A new family of parametric algebraic activation (PAA) functions has been proposed in this paper for multi-layered feedforward neural networks. The proposed PAA is S-shaped for various values of the parameter. Elliott function is a special case for the PAA. It has been proved that the proposed PAA is well-defined, continuous and bounded. It has been shown that PAA is a smooth and increasing function and the range of PAA belongs to $[-1\ 1]$. PAA is designed in such a fashion that its slope at the origin is one. The proposed PAA converges to sign function as the parameter tends to infinity. Resilient propagation algorithms (RPROPs) [35] have been used to test the performance evaluation of PAA and comparison has been made with other algebraic and well-known transcendental activation functions on various benchmark datasets from University of California Irvine (UCI) machine learning repository.

Rest of the paper is organized as follows. Section 2 presents overview of activation functions. Section 3 gives the details of proposed PAA function and its properties. Section 4 describes weight updating equations of resilient propagation using PAA. Section 5 gives the comparative performance evaluation of the proposed PAA on various benchmark datasets taken from UCI machine learning repository. Conclusions are given in Section 6.

2. OVERVIEW OF ACTIVATION FUNCTIONS

Activation functions that exist in the literature available thus far can be grouped into transcendental and algebraic activation functions. The well-known activation functions like sigmoid and tan-hyperbolic are transcendental activation functions. Step function, piecewise linear function, and algebraic sigmoid function are algebraic activation functions. Piecewise linear functions are suitable only for single layer perceptron and they are not differentiable. The existing activation functions are briefly discussed below:

2.1 Sigmoid Activation Function

Sigmoid function [4] is the most standard form of activation function used in the field of ANN. An example of the sigmoid function is the logistic function. It is bounded, differentiable, and attains slope value of one at the origin and it has the range $[0, 1]$. It is given by

$$\varphi\ (x) = \frac{1}{1\ + \text{o}e^{-x}} \tag{1}$$

2.2 Tan-hyperbolic Activation Function

This activation function ranges from -1 to $+1$ and assumes an anti-symmetric form with respect to origin. Tan-hyperbolic function [5] is defined as

$$\varphi\ (x) = \frac{e^x\ -\ e^{-x}}{e^x\ +\ e^{-x}} \tag{2}$$

2.3 Piecewise Linear Function

Piecewise linear function [5] is defined on $[0\ 1]$ as shown below. It is only continuous but not differentiable at 0.5

and -0.5.

$$\varphi\ (x) = \begin{cases} 1\ x \geq \dfrac{1}{2} \\ x\ +\ \dfrac{1}{2}\ \ -\ \dfrac{1}{2} \leq x \leq \dfrac{1}{2} \\ 0\ x \leq\ -\dfrac{1}{2} \end{cases} \tag{3}$$

2.4 Algebraic Sigmoid Function

Algebraic sigmoid function [29] is defined as

$$\varphi\ (x) = \frac{x}{\sqrt{1\ +\ x^2}} \tag{4}$$

It is well-defined as the denominator does not vanish on R. It is denoted as AF1 in the rest of the paper.

2.5 Elliott Function

Another algebraic function, Elliott function [26] is defined as

$$\varphi\ (x) = \frac{x}{1\ +\ |x|} \tag{5}$$

It is well-defined as $1 + |x| \neq 0$ on R. It is continuous, differentiable, bounded and attains slope one at the origin.

2.6 Parametric Sigmoid Function

Parametric sigmoid function (PSF) [11] is continuous, differentiable, and bounded. It is defined as

$$\varphi\ (x) = \left(\frac{1}{1\ +\ e^{-x}}\right)^m \quad \text{for all } m \quad (0,\ \infty\ ) \tag{6}$$

3. PROPOSED WORK

3.1 Parametric Algebraic Activation (PAA)

In the available literature, various transcendental activation functions like sigmoid and tan-hyperbolic have been used. Very few algebraic activations have been developed of which Elliott function is one of them. None of the differentiable algebraic activation functions developed so far contains a parameter which converges to sign function as the parameter tends to infinity. This motivated us to develop new PAA function of which Elliott function is a special case. It satisfies all the important properties that an activation function should possess. The proposed PAA function is given by

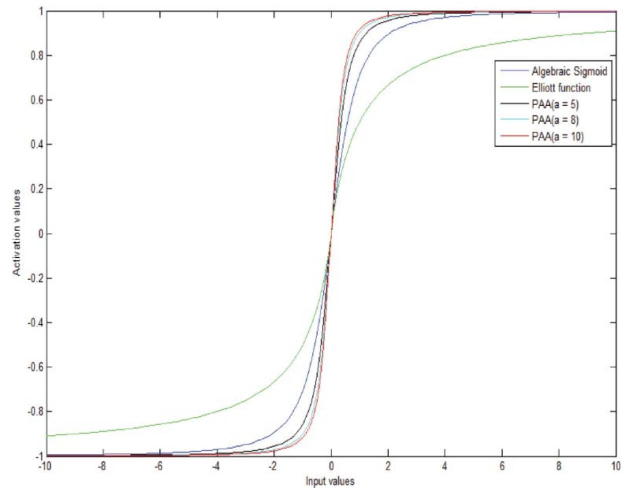

Figure 1: Graph of the various activation functions

$$f_a\ (x) = \frac{x(1\ +\ a{*}|x|)}{1\ +\ |x|{*}(1\ +\ a{*}|x|)} \quad \text{For } x\,\varepsilon\, R \text{ and } a \geq 0 \tag{7}$$

The function $f_a(x)$ denotes a family of S-shaped curves for various values of parameter a. Substituting $a = 0$ in proposed function shown in Equation (7) gives the well-known Elliott function. Figure 1 shows the graphical representation of algebraic sigmoid, Elliott, and PAA for $a = 5$, 8, and 10. All the activation functions of a neural network possess a slope value of one at the origin and satisfy well-defined property. Proposed activation function not only attains slope one at the origin but also well-defined, bounded, continuous, and differentiable on R. The first neuron model by McCulloch–Pitts [36] uses sign function as an activation function which is linear activation function. The PAA converges to sign function when a tends to infinity. The range of $f_a(x)$ belongs to $[-1\ 1]$. In order to change range of output neuron from $[-1\ 1]$ to $[0\ 1]$, $[(1 + f_a)/2]$ is used instead of f_a. Proofs for the above-mentioned properties are discussed in detail as follows.

Theorem 3.1: $f_a(x)$ is well-defined on R for a ≥ 0.

Proof: Function f is well-defined if and only if

$$x = y \text{ implies } f_a(x) = f_a(y)\ \forall\ a \geq 0$$

Now we are going to prove denominator of (f_a) is non-zero on R.

Let the denominator of the proposed function (f_a) defined in Equation (9) be denoted by

$$Q(x) = 1 + |x|{*}(1 + a{*}|x|)$$

$$Q(x) = 1 + |x| + a{*}\mathrm{x}^2$$

We know that $\forall a \geq 0$ and $\forall x \, \varepsilon \, R$

$$|x| \geq 0 \text{ and } a * x^2 \geq 0 => 1 + |x| + a * x^2 \geq 1 \\ => Q(x) \geq 1$$

Hence it is proved that $Q(x)$ is not equal to zero on R

Consider:

$$x = y => x\,(1 + a*|x|) = y(1 + a*|y|)$$

Since Q is nonzero, divide L.H.S and R.H.S of above equation by Q

$$=> \left(\frac{x(1+a*|x|)}{Q(x)}\right) = \left(\frac{y(1+a*|y|)}{Q(y)}\right) => fa(x) = fa(y)$$

Since $Q(x)$ is nonzero on real numbers.

Hence, it is proved that $f_a(x)$ is well-defined on R for $a \geq 0$.

Theorem 3.2: The family of proposed activation functions $(f_a(x))$ converges to sign function, i.e. for all $\varepsilon > 0$ there exists c such that $\left|f_a(x) - \frac{x}{|x|}\right| < \varepsilon$ for $a > c$.

Proof: This is proved for two cases:

Case 1.

$$\text{Let } x \, \varepsilon \, R,\ x > 0 \text{ and } |f_a(x) - 1| < \varepsilon \tag{8}$$

Substituting $f_a(x)$ from Equation (7) which is given by

$$\left|\frac{x(1+a|x|)}{1+|x|(1+a|x|)} - 1\right| < \varepsilon \tag{9}$$

The value of a obtained from Equation (9) is given by

$$\left(\frac{1}{\varepsilon} - 1 - x\right)\frac{1}{x^2} < a \tag{10}$$

From Equation (10), the value of c is to be taken as

$$c = \left(\frac{1}{\varepsilon} - 1 - x\right)\frac{1}{x^2} \tag{11}$$

for any given $x \, \varepsilon \, R$ and $x > 0$.

Hence, it has been proved that $x \, \varepsilon \, R$ and $x > 0$, $\forall \varepsilon > 0 \ \exists c \ni \forall a > c: \ |f_a(x) - 1| < \varepsilon$

Case 2. Consider $x \, \varepsilon \, R,\ x < 0$ and $|f_a(x) + 1| < \varepsilon$.
Substituting $f_a(x)$ from Equation (7) which is given by

$$\left|\frac{x(1+a|x|)}{1+|x|(1+a|x|)} + 1\right| < \varepsilon \tag{12}$$

The value of a obtained from Equation (12) is given by

$$\left(\frac{1}{\varepsilon} - 1 + x\right)\frac{1}{x^2} < a \tag{13}$$

From the above equation, the value of c is to be taken as

$$c = \left(\frac{1}{\varepsilon} - 1 + x\right)\frac{1}{x^2} \tag{14}$$

for any given $x \, \varepsilon \, R$ and $x < 0$

$x \, \varepsilon \, R$ and $x < 0, \forall \varepsilon > 0 \ \exists c \ni \forall a > c: \ |f_a(x) + 1| < \varepsilon$

Case 3. Consider $x = 0$, we get $f_a(0) = 0 \ \forall a$

For $x = 0, \forall \varepsilon > 0 \ \exists c \ni \forall a > c: \ |f_a(x)| < \varepsilon$
Hence from the above three cases, it is concluded that proposed function $f_a(\mathrm{x})$ approaches to sign function as a goes to infinity.

The following lemma is required to prove that proposed activation function is increasing.

Lemma 3.1: Family of proposed activation functions $(f_a(x))$ is differentiable functions on R.

Proof: At $x = 0$, the differentiability can be obtained as $f_a'(0+h) = 1, f_a'(0-h) = 1$ as $h \to 0$ and $f_a'(0) = 1$. This shows that the derivative of $f_a(x)$ exists at zero also. Hence $f_a(x)$ is differentiable on R for $a \geq 0$.

Lemma 3.2: Family of activation functions $f_a(x)$ is continuous functions on R for $a \geq 0$.
Proof: $f_a(x)$ is differentiable on R for $a \geq 0$ implies that family of activation functions $f_a(x)$ is continuous functions on R for $a \geq 0$.
Note: It is seen from Equation (14) that c depends on the choice of x and ε and hence the family of functions $(f_a(x))$ converges pointwise to the Heaviside step function. Since Heaviside step function is a discontinuous function, by uniform limit theorem, we can conclude that the family of functions $(f_a(x))$ does not converge uniformly to the Heaviside step function.

Theorem 3.3: Family of activation functions $(f_a(x))$ is increasing function on R.

Proof: Using chain rule of differentiation, the derivative $f_a'(x)$ of $f_a(x)$ is given by

$$f_a'(x) = \frac{1 + 2a*|x|}{[1 + |x|*(1 + a*|x|)]^2} \tag{15}$$

It is clear from above equation that $f_a'(x) \geq 0 \ \forall x$. It is concluded that $f_a(x)$ is increasing on R.

Theorem 3.4: Family of proposed activation functions $(f_a(x))$ is bounded functions on R

***Proof*:** It is known that a continuous real function with finite limits is bounded. From Lemma 3.2, family of activation functions $f_a(x)$ is continuous functions on R for $a \geq 0$. It is seen that $\lim_{x \to \pm\infty} f_a(x) = \pm 1$. Hence, family of proposed activation functions $(f_a(x))$ is bounded functions on R.

Proposed activation function is applied on resilient propagation neural network. The following section deals with the weight updating equations using resilient propagation.

4. RESILIENT PROPAGATION USING PAA

Resilient propagation (R-prop) [35] is used as a learning technique for supervised learning in feedforward ANN using PAA as activation function. Steepest descent is used for minimizing the error function. Figure 2 shows the neural network architecture with n input neurons, m hidden neurons, and k output neurons.

Let the total error E be denoted by

$$E = \frac{1}{2}\sum_r (t_r - \mathrm{out}_r)^2 \tag{16}$$

where t_r denotes target and out_r denotes output of the activation at the rth hidden neuron.

Let W_{ij} be the weight connection between ith input neuron to jth hidden neuron.

Let V_{jr} be the weight connection between jth hidden neuron to rth output neuron.

Let PH_{ij} denotes partial derivative of E with respect to W_{ij} and PO_{jr} denotes partial derivative of E with respect to V_{jr}.

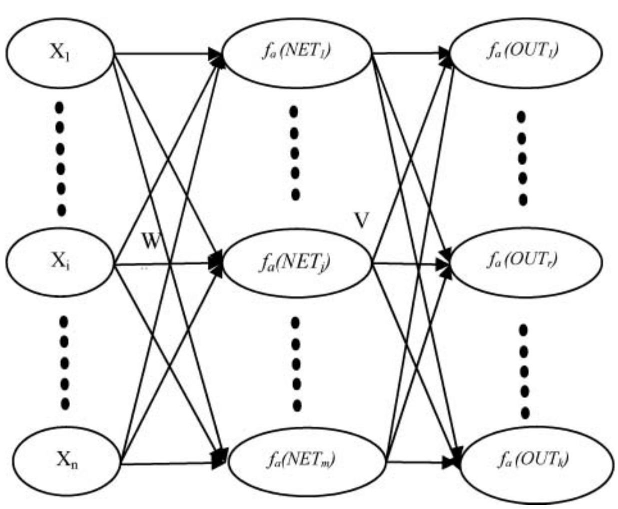

Figure 2: Neural network architecture for resilient propagation

The partial derivative of error (E) with respect to the weight (V_{jr}) is given by

$$\mathrm{PO}_{jr} = -(t_r - \mathrm{out}_r)*\frac{1+2*a*|\mathrm{net}_r|}{(\mathrm{net}_r)^2(1+a*|\mathrm{net}_r|)^2}*(\mathrm{out}_r)^2*(\mathrm{out}_j) \tag{17}$$

Error at the output node is denoted by δ_r, which is given by

$$\delta_r = (t_r - \mathrm{out}_r)*\frac{1+2*a*|\mathrm{net}_r|}{(\mathrm{net}_r)^2(1+a*|\mathrm{net}_r|)^2}*(\mathrm{out}_r)^2 \tag{18}$$

Similarly, the partial derivative of error (E) with respect to W_{ij} using steepest descent principle is given by

$$\mathrm{PH}_{ij} = -\frac{1+2*a*|\mathrm{net}_j|}{(\mathrm{net}_j)^2(1+a*|\mathrm{net}_j|)^2}.(\mathrm{out}_j)^2\sum_r \delta_r V_{jr} x_i \tag{19}$$

The weight change is obtained by the following equation:

$$\Delta W_{ij}(t) = \begin{cases} -\emptyset_{ij}(t) \ \textit{if} \ \mathrm{PH}_{ij}(t) > 0 \\ \emptyset_{ij}(t) \ \textit{if} \ \mathrm{PH}_{ij}(t) < 0 \\ 0 \ \textit{otherwise} \end{cases} \tag{20}$$

where

$$\emptyset_{ij}(t) = \begin{cases} \beta^+\emptyset_{ij}(t-1) \ \text{if} \ \mathrm{PH}_{ij}(t-1)*\mathrm{PH}_{ij}(\mathrm{t}) > 0 \\ \beta^-\emptyset_{ij}(t-1) \ \text{if} \ \mathrm{PH}_{ij}(t-1)*\mathrm{PH}_{ij}(\mathrm{t}) < 0 \\ \emptyset_{ij}(t-1) \ \text{otherwise} \end{cases}$$

where $0 < \beta^+ < 1 < \beta^-$.

The weight updating equation is given by

$$W_{ij}^{(t+1)} = W_{ij}^{(t)} + \Delta W_{ij}^{(t)} \tag{21}$$

$$V_{jr}^{(t+1)} = V_{jr}^{(t)} + \Delta V_{jr}^{(t)} \tag{22}$$

Performance evaluation of PAA is given in the results section. Apart from using the efficient activation function, the choice of the right number of hidden neurons also plays an important role to improve the classification accuracy in neural network. The right choice of number of hidden neurons in a multi-layered feedforward neural network helps in deciding the architecture of the neural network. Reduction in number of hidden neurons also reduces the computational time.

Table 1: List of datasets used for classification

S.No	Dataset name	Features	Classes
1	Breast cancer	9	2
2	Haberman	3	2
3	Iris	4	3
4	Ecoli	8	8
5	Pima Indian diabetes	8	2
6	Mammographic	6	2
7	Seeds	7	3
8	Skin segmentation	4	2
9	Planning relax	13	2
10	Image segmentation	19	7
11	Yeast	8	10
12	Indian liver patient dataset (ILPD)	10	2
13	Fertility diagnosis	10	2
14	Glass identification	9	7
15	Phoneme	5	2
16	Texture	40	11
17	Satimage	36	7

5. EXPERIMENTAL ANALYSIS

Performance evaluation of PAA was done with RPROP on various benchmark datasets. Comparison of performance was done using widely known activation functions like the sigmoid, PSF($m = 5$), tan-hyperbolic, inverse tan, Elliott algebraic activation function, and other algebraic activation functions. Datasets were taken from UCI machine learning repository and Elena project real datasets. Nomenclature of the datasets is given in Table 1. Ten cross validation was performed on all the datasets and the results depict the average number of epochs. Average number of epochs at which the validation error starts to increase was observed using resilient propagation and these are given in Table 2. For most of the datasets, the average number of epochs is reduced using PAA when compared with other activation functions. The reduction in the number of epochs for PAA is drastic when compared to available algebraic activation function AF1. Testing errors for all the datasets corresponding to different activation functions are shown in Table 3. For the comparison of training epochs and testing errors attained using PAA and other activation functions, pair wise t-test was performed and the corresponding p-values are given in Tables 4 and 5. It is observed from Tables 4 and 5 that the reduction in the

Table 2: Average number of training epochs using resilient propagation

S. No	Proposed function ($a = 5$)	Sigmoid	AF1	Tan$h$ (x)	Tan^{-1}(x)	Elliot	PSF ($m = 5$)
1	7.7	12.4	14.2	11.3	8.5	13.2	10.5
2	3.5	4.2	9.8	13.5	5.3	5.7	5.2
3	5.7	6.4	6.4	7.4	24.2	25.7	7.8
4	5.8	7.1	13.5	11.5	26.6	6.4	6
5	4.8	5.5	5.8	7.0	6.3	5.7	5.5
6	5.5	11	11.5	8.7	4.9	5.7	9.4
7	5.25	6.98	9.2	8.6	7.3	8	7.82
8	6.28	8.3	10.4	7.5	15	9	7.92
9	2.8	10.5	31.1	21.1	15.4	7.2	8.6
10	4.7	15.6	10.4	7.2	11.4	8.3	13.4
11	13.5	17.5	25.6	15.2	14.7	7.4	14.6
12	9.5	13.5	30.9	12.6	25.2	11.3	15.2
13	2	13.8	16.8	13.2	5	4.6	8.5
14	7.5	5.8	12.5	12.7	9.3	8.2	9.2
15	9.3	15.63	16.67	14.43	12.63	16.42	12.43
16	13.5	15.3	15.5	18.3	15.64	17.8	16.2
17	10.2	12.12	16.43	13.42	14.5	13.42	13.2

Table 3: Testing error for various activations using resilient propagation

S. No	Proposed function ($a = 5$)	Sigmoid	AF1	Tan$h$(x)	Tan^{-1}(x)	Elliot	PSF ($m = 5$)
1	0.0025	0.0032	0.0070	0.0071	0.0062	0.0041	0.0029
2	0.01483	0.01683	0.04216	0.01816	0.02245	0.01328	0.01921
3	0.01994	0.02432	0.04030	0.02143	0.01974	0.0231	0.02213
4	0.01297	0.01316	0.02769	0.01749	0.01300	0.1194	0.01621
5	0.01692	0.02745	0.03297	0.04523	0.01832	0.01819	0.02238
6	0.02246	0.0431	0.2432	0.0223	0.02438	0.02452	0.0321
7	0.10427	0.2042	0.4526	0.342	0.3214	0.2043	0.1542
8	0.01621	0.02042	0.0173	0.0134	0.0809	0.0890	0.01947
9	0.1187	0.1342	0.3246	0.3527	0.1359	0.1445	0.1445
10	0.0985	0.1273	0.1456	0.1144	0.1327	0.1322	0.1113
11	0.1998	0.2140	0.3181	0.2211	0.2123	0.3212	0.2074
12	0.01432	0.02145	0.02658	0.02007	0.01995	0.01556	0.0198
13	0.1055	0.2142	0.3850	0.2003	0.14623	0.1824	0.2132
14	0.01326	0.02325	0.02785	0.02137	0.0952	0.0333	0.01892
15	0.01137	0.09998	0.0496	0.0364	0.03599	0.0349	0.06231
16	0.09001	0.3945	0.5546	0.5727	0.3554	0.3999	0.35313
17	0.01994	0.09953	0.04945	0.06162	0.0489	0.0562	0.0753

Table 4: p-Values for training epochs using resilient propagation

Activation function	Sigmoid	AF1	Tan$h$(x)	Tan^{-1}(x)	Elliot function	PSF
p-Values	0.0034	0.00014	0.00018	0.0013	0.0214	0.0054

Table 5: p-Values for training errors using resilient propagation

Activation function	Sigmoid	AF1	Tan$h$ (x)	Tan^{-1}(x)	Elliot function	PSF
p-Values	0.0589	0.0112	0.0497	0.0587	0.0433	0.0750

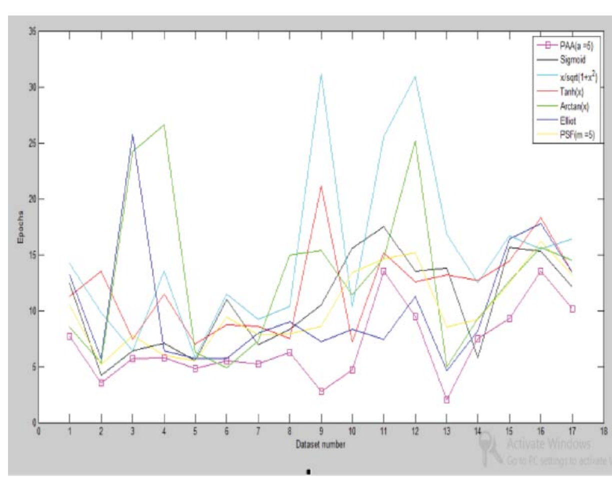

Figure 3: Number of training epochs achieved using resilient propagation with various activation functions

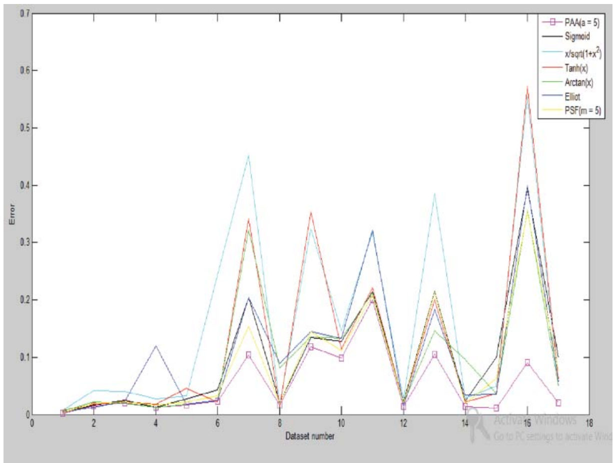

Figure 4: Testing error using resilient propagation for various activations

epochs and testing errors using PAA are significant at the level of significance 0.05.

Line graph representation of training epochs and testing errors for all the datasets are shown in Figures 3 and 4, respectively. It can be inferred from the graphs that number of epochs and testing errors are reduced using PAA in comparison with other activation functions.

6. CONCLUSION

Novel PAA function has been proposed in this paper for internal adjustment in a multi-layer feedforward neural network. The advantage of this activation function is that it is general in nature, converges to linear activation function, i.e. sign function as the parameter tends to infinity. Elliot algebraic activation function is a special case of this function. Resilient propagation has been used for training. The superiority of performance of PAA has been demonstrated on various benchmark datasets by comparing with other well-known activation functions. The reduction in the number of epochs, testing errors for PAA in RPROP is drastic and statistically significant when compared to other activation functions.

ACKNOWLEDGMENTS

We are thankful to the director Dr G.R.C. Reddy, NITGOA for providing constant encouragement throughout this research work. We are also thankful to reviewers for their valuable suggestions.

DISCLOSURE STATEMENT

No potential conflict of interest was reported by the authors.

REFERENCES

1. S. Goh and D. P. Mandic, "Recurrent neural networks with trainable amplitude of activation functions," *Neural Netw.*, Vol. 16, no. 8, pp. 1095–1100, May 2003.
2. M. Solazzi and A. Uncini, "Regularising neural networks using flexible multivariate activation function," *Neural Netw.*, Vol. 17, no. 2, pp. 247–60, Mar. 2004.
3. L. Vecci, F. Piazza, and A. Uncini, "Learning and approximation capabilities of adaptive spline activation function neural networks," *Neural Netw.*, Vol. 11, no. 2, pp. 259–70, Mar. 1998.
4. J. Han and C. Morag, "The influence of the sigmoid function parameters on the speed of back propagation learning," in *Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*, Spain, 1995, pp. 195–201.
5. S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Singapore: Pearson Education, 1998.
6. Y. Pao, H, *Adaptive Pattern Recognition and Neural Networks*, Reading, MA: Addison-Wesley, 1989, pp. 161–70.
7. D. von Seggern, *CRC Standard Curves and Surfaces with Mathematics*, 2nd ed. Boca Raton, FL: CRC Press, 2007.
8. T. M. Mitchell, *Machine Learning*, Hoboken, NJ: John Wiley & Sons, 1997.
9. D. Hosmer and S. Lemeshov, *Applied Logistic Regression*, NewYork: Wiley & Sons, 1989.
10. J. Dombi, "Membership function as an evaluation," *Fuzzy Sets Syst.*, Vol. 35, pp. 1–22, 1990, doi: 10.1016/0165-0114(90)90014-W.
11. P. Chandra and Y. Singh, "An activation function adapting training algorithm for sigmoidal feedforward networks," *Neurocomput. Lett.*, Vol. 61, pp. 429–37, Oct. 2004.
12. L. Zhou and L. Zhang, "A log-sigmoid Lagrangian neural network for solving nonlinear programming," in *Proceedings of ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Qingdao, China, Jul. 30–Aug. 1, 2007, pp. 427–31.
13. G. A. Anastassiou, *Intelligent Systems: Approximation by Artificial Neural Networks, Intelligent Systems Reference Library 19*. Berlin: Springer-Verlag, 2011.
14. A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inform. Theory*, Vol. 39, no. 3, pp. 930–45, May 1993.
15. G. Lewicki and G. Marino, "Approximation of functions of finite variation by superpositions of a sigmoidal

function," *Appl. Math. Lett.*, Vol. 17, pp. 1147–52, Jan. 2004.

16. Z. Chen and F. Cao, "The approximation operators with sigmoidal functions," *Comput. Math. Appl.*, Vol. 58, no. 4, pp. 758–65, Aug. 2009.
17. Z. Chen and F. Cao, "The construction and approximation of a class of neural networks operators with ramp functions," *J. Comput. Anal. Appl.*, Vol. 14, no. 1, pp. 101–12, Jan. 2012.
18. G. H. L. Cheang, "Approximation with neural networks activated by ramp sigmoids," *J. Approx. Theory*, Vol. 162, pp. 1450–65, Aug. 2010.
19. D. Costarelli and R. Spigler, "Approximation by series of sigmoidal functions with applications to neural networks," *Annali di Matematica Puraed Appl.*, Vol. 194, no. 1, pp. 289–306, Feb. 2015.
20. D. Costarelli and V. Vinti, "Approximation by max-product neural network operators of Kantorovich type," *Results Math.*, Vol. 69, no. 3, pp. 505–19, Jun. 2016.
21. N. Hahm and B. Hong, "Approximation order to a function in C(R) by superposition of a sigmoidal function," *Appl. Math. Lett.*, Vol. 15, pp. 591–7, Jul. 2002.
22. N. Kyurkchiev and S. Markov, *Sigmoid Functions: Some Approximation and Modelling Aspects, Some Moduli in Programming Environment Mathematica*, UK: LAP (Lambert Acad. Publ.), 2015.
23. D. Costarelli and R. Spigler, "Solving Volterra integral equations of the second kind by sigmoidal functions approximation," *J. Int. Equ. Appl.*, Vol. 25, no. 2, pp. 193–222, Nov. 2013.
24. D. Costarelli and R. Spigler, "A collocation method for solving nonlinear Volterra integro-dierential equations of the neutral type by sigmoidal functions," *J. Int. Equ. Appl.*, Vol. 26, no. 1, pp. 15–52, Nov. 2014.
25. D. Costarelli, "Neural network operators: Constructive interpolation of multivariate functions," *Neural Netw.*, Vol. 67, pp. 28–36, Jul. 2015.
26. D. Elliot, "A better activation function for artificial neural networks, the National Science Foundation," Institute for Systems Research, Washington, DC, ISR Technical Rep. TR-8, 1993. Available: http://ufnalski.edu.pl/zne/ci_2014/papers/Elliott_TR_93-8.pdf
27. B. Mendil and K. Benmahammed, "Simple activation functions for neural and fuzzy neural networks," in *Proceedings of IEEE International Symposium on Circuits and Systems*, Orlando, FL, May 30–Jun. 2, 1999, pp. 347–50.
28. J. Sopena, E. Romero, and R. Alquézar, "Neural networks with periodic and monotonic activation functions: a comparative study in classification problems," in *Proceedings of International Conference on Artificial Neural Networks*, 1999, pp. 323–8.
29. W. Duch and N. Jankowski, "Survey of neural transfer functions. *Neural Comput. Surveys*, Vol. 2, pp. 163–212, 1999.
30. J. Kamnuzaman and S. Aziz, "A note on activation function in multilayer feedforward learning," in *Proceedings of International Joint Conference on Neural Networks*, Honolulu, Hawaii, May 12–17, 2002, pp. 519–23.
31. G. S. D. S. Gomes and T. B. Ludermir, "Optimization of the weights and asymmetric activation function family of neural network for time series forecasting," *Expert Syst. Appl.*, Vol. 40, no. 16, pp. 6438–46, Nov. 2013.
32. G. Miguez, A. E. Xavier, and N. Maculan, "An evaluation of the bihyperbolic function in the optimization of the backpropagation algorithm," *Int. Trans. Operat. Res.*, Vol. 21, no. 5, pp. 835–54, Sep. 2014.
33. M. Abdulkarim, W. F. W. Ahmad, A. Shafie, and R. Razali, "Performance of multi-layer perceptron neural networks with an exponential decay activation function in airwaves estimation," *Int. J. Adv. Studies Comput. Sci. Eng.*, Vol. 2, no. 2, pp. 25, 2013.
34. S. M. Siniscalchi and V. M. Salerno, "Adaptation to new microphones using artificial neural networks with trainable activation functions," *IEEE Trans. Neural Netw. Learn. Syst.*, Apr. 2016.
35. M. Riedmiller and H. Braun, "A direct adaptive method for faster back-propagation learning: The RPROP algorithm," in *Proceedings of the International Conference on Neural Networks, San Francisco*, San Francisco, CA, Mar. 28–Apr. 1, 1993, pp. 586–91.
36. W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, Vol. 5, pp. 115–33, 1943.

Authors

K. V. Naresh Babu is presently working as an assistant professor in Computer Science and Engineering Department at NIT Goa. He has successfully graduated with MTech and PhD degrees from Indian Institute of Technology Delhi. He has also worked as a senior research associate at Evalueserve Pvt Ltd. His research interests include big data analytics and machine learning. He has published several research publications in national and international conferences and journals.

E-mail: venkatanaresh@nitgoa.ac.in

Damodar Reddy Edla received BSc degree from Kakatiya University in 2004, MSc degree from University of Hyderabad in 2006 and MTech and PhD degrees in Computer Science and Engineering from Indian School of Mines, Dhanbad in 2009 and 2013, respectively. His research area is data mining and wireless sensor networks. He is currently Head of the Computer Science and Engineering Department, National Institute of Technology, Goa, India. He has more than 20 research publications in reputed international journals and conferences.

E-mail: dr.reddy@nitgoa.ac.in