

A Low-Complexity Fuzzy Activation Function for Artificial Neural Networks

Emilio Soria-Olivas, José D. Martín-Guerrero, Gustavo Camps-Valls, Antonio J. Serrano-López, Javier Calpe-Maravilla, and Luis Gómez-Chova

Abstract—A novel fuzzy-based activation function for artificial neural networks is proposed. This approach provides easy hardware implementation and straightforward interpretability in the basis of IF-THEN rules. Backpropagation learning with the new activation function also has low computational complexity. Several application examples (XOR gate, chaotic time-series prediction, channel equalization, and independent component analysis) support the potential of the proposed scheme.

Index Terms—Activation function, fuzzy logic, rule extraction.

I. INTRODUCTION

The multilayer perceptron is one of the most popular artificial neural networks (ANNs) because of the universal approximation theorem [1]. This network is composed of a series of elements, neurons, arranged in layers and interconnected through synaptic weights. The output of the neuron takes a nonlinear function of the weighted sum of its inputs.

The first activation function proposed was the sign function by McCulloch-Pitts. Since this function is not differentiable, smoother functions such as the sigmoid (output range between 0 and 1) and the hyperbolic tangent (output range between ± 1) are preferred for analytical convenience. As these are nonlinear functions, they are difficult to implement in hardware. A possible approach to circumvent this problem is using piecewise-linear functions, but the resulting activation function suffers from nondifferentiability at the intersections between the different linear functions. A second possibility is to make use of Taylor expansion and withdraw all but the first (linear) terms. However, with this approach the main characteristic of a neural network (its nonlinearity) is lost. In [2], another approach was heuristically proposed to provide a simple sigmoid-like nonlinear activation function more suitable for digital hardware implementation

$$f(x) = \begin{cases} \text{sign}(x), & |x| \geq L \\ -\frac{x|x|}{L^2} + \frac{2x}{L}, & \text{otherwise.} \end{cases} \quad (1)$$

In the present paper, we show that the activation function of (1) can be drawn in a more natural way by defining the classical activation function by means of the fuzzy logic methodology. This, in turn, provides some advantages, such as a straightforward interpretation of the results obtained by using the IF-THEN rules embedded in the ANN, a low computational burden is achieved since weight updating is not always necessary, and an easy learning algorithm is reproduced. The methodology followed here is generally applicable to any existing activation function, which does not preclude its use in neural networks with nonsigmoidal activation functions such as radial basis function (RBF) neural networks.

This paper is organized as follows. Section II presents the theoretical development and learning rule of the new activation function proposed. Section III includes four application examples showing the advantages of our proposal: the XOR-gate problem; chaotic time-series prediction;

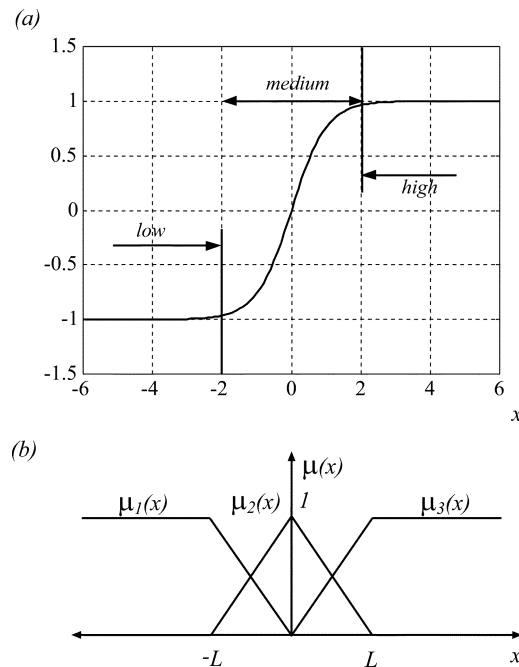


Fig. 1. (a) Linguistic variables used. The saturated zones of the activation function identified with *high* and *low* labels. The sigmoidal tract has the *medium* linguistic variable. (b) Schematic of three membership functions used to describe the linguistic variables depicted in (a).

channel equalization; and independent component analysis (ICA). Section IV is devoted to IF-THEN rule generation from the saturation zones of a trained perceptron. Finally, Section V offers some conclusions and outlines for further work.

II. THEORETICAL DEVELOPMENT

The hyperbolic tangent activation function is modeled by means of linguistic variables as shown in Fig. 1 and described by

$$f(x) = \begin{cases} -1, & x \text{ is low} \\ a \cdot x, & x \text{ is medium} \\ +1, & x \text{ is high} \end{cases} \quad (2)$$

where a is a constant factor representing the smoothness of the sigmoid tract [see Fig. 1(a)]. In the context of fuzzy logic, x can be regarded as to a linguistic variable, which can be defined by a series of membership functions. We will consider here triangular functions due to their simplicity [3], as depicted in Fig. 1(b). The membership functions refer to the *low*, *medium*, and *high* concepts, respectively. Finer partitions could be examined with fuzzy numbers such as *very low*, *low*, *medium*, *high*, and *very high*. This would improve the modeling of the activation function, but it would also increase the computational burden.

The value of the function at a specific point x_o is given by

$$f(x_o) = (-1)\mu_1(x_o) + ax_o\mu_2(x_o) + (+1)\mu_3(x_o). \quad (3)$$

Since the activation function must be continuous, relation $a = 1/L$ must be satisfied and, thus, expression (1) can be readily obtained from (2) and (3). Several characteristics about the resulting function can be stated as follows:

- The function $f(x)$ does not correspond to the Taylor series expansion of the activation function, since the second derivative of the hyperbolic tangent is zero at the origin. Therefore, no quadratic term appears in this expansion.
- The function is differentiable at every point in its entire domain.

Manuscript received October 23, 2002; revised December 2, 2002.

The authors are with the Group de Processament Digital de Senyals, Department of Enginyeria Electrònica, Universitat de València, València 46100, Spain (e-mail: soriae@uv.es).

Digital Object Identifier 10.1109/TNN.2003.820444

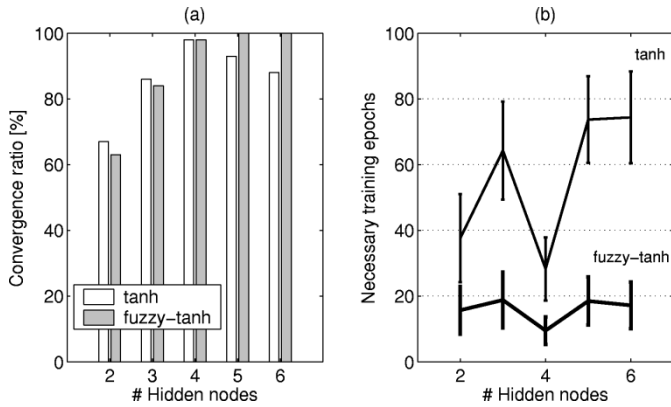


Fig. 2. Outcomes for the XOR-gate problem using the standard hyperbolic tangent (tanh) and the fuzzy tanh proposed. (a) Convergence rate. (b) Mean \pm standard deviation of the number of epochs required by networks to solve the problem, as a function of the number of hidden nodes.

- The membership functions defined in Fig. 1(b) are easily realizable in hardware by using threshold comparators and simplify the implementation of the activation function in an electronic device, such as a microcontroller, digital signal processor, or field-programmable gate array.
- The way of defining the domain of variable x by means of linguistic variables allows easy interpretation of the result provided by the neural network based on IF-THEN rules.

Note that we must force weight update if there is an error in the output layer and x lies in the saturation zone. This problem can be avoided by including a simple condition in the delta rule: whenever the derivative of the activation function is zero and there is an output error, the value of the derivative is set to a constant value, for simplicity $+1$.

III. EXPERIMENTAL RESULTS

We use four problems in ANNs to verify the performance of the proposed activation function: implementation of the XOR gate, prediction of the Mackey–Glass time series, and equalization of communication channels and ICA. Two ANNs with identical architecture and initial weights were considered in every problem. One uses the hyperbolic tangent and the other uses the proposed activation function with the modified learning algorithm. We used the same learning rates and momentum terms; the derivative of the hyperbolic tangent in the output was removed in order to avoid saturation problems.

A. Problem of the XOR Gate

In this preliminary test, we evaluate convergence issues of ANNs using our proposed method. The aim is to model the logic of a two-input XOR gate by means of ANNs. We verify the speed of convergence of the two models by fixing a threshold value of 0.1 for the addition of the square errors of the four patterns. Fig. 2(a) shows the average (1000 runs) of ANNs that converge. Fig. 2(b) depicts the number of epochs required to reach the threshold. The application of the new activation function and the modification of the backpropagation algorithm both raise the convergence rate and training speed of ANNs [Fig. 2(b)]. Moreover, the proposed activation function induces a lower number of operations for training the ANNs and, thus, an improved execution time.

TABLE I
OUTCOMES FOR THE PREDICTION OF THE
MACKEY–GLASS TIME SERIES. RESULTS USING THE
FUZZY ACTIVATION FUNCTION ARE SHOWN IN A BOLD FONT

Hidden nodes	5	6	7	8	9
Mean Absolute	0.057	0.103	0.089	0.073	0.060
Error	0.027	0.039	0.063	0.035	0.036
Std Absolute	0.054	0.063	0.093	0.054	0.055
Error	0.062	0.034	0.082	0.086	0.045

B. Prediction of the Mackey–Glass Time Series

The one-step-ahead forecasting capabilities of our proposal are tested using the classical high-dimensional chaotic system generated by the Mackey–Glass delay differential equation

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-t_\Delta)}{1+x(t-t_\Delta)^{10}} \quad (4)$$

with delay $t_\Delta = 17$. We consider 1000 training samples and use the next 500 for free parameter selection (validation set), as proposed in [4].

The absolute value of the prediction error and its variance are used as working indexes. Table I shows the results obtained for different number of hidden nodes by fixing the number of network inputs (embedding dimension). The joint new-activation function and the proposed modification of the delta rule proved to be a good alternative to the hyperbolic tangent.

C. Channel Equalization

When a binary signal is transmitted through a real dispersive channel, the received signal is affected by intersymbol interference. Moreover, if noise is present, further corruption ensues. Therefore, in many practical cases, equalization is necessary to recover the information from the received signal. Under adverse conditions such as low signal-to-noise ratio or when the distribution of the received samples is not linearly separable, nonlinear techniques are preferred and ANNs are becoming a common choice [5].

In this application example, the original sequence is considered binary and the channel output is affected by white noise. Two channels in Z-domain form are considered, as follows:

$$H_1(z) = 0.5 + z^{-1} \quad (5)$$

and

$$H_2(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}. \quad (6)$$

Simulations consider a training set containing 250 elements and 250 more samples for validation. Additive Gaussian noise with zero mean and variance 0.3 is introduced. We evaluate the convergence rate as the percentage of networks with success rates in the validation set higher than 75% after 500 epochs. As in the first application, ANNs use the same weight initialization and learning rates.

Fig. 3 shows both the convergence ratio and the percentage of weight updates when using the fuzzy approach for $H_1(z)$ [Fig. 3(a) and (b)] and $H_2(z)$ [Fig. 3(c) and (d)]. Results are averaged over 100 realizations for each network. Convergence is ensured by using our proposal and performance is similar to the usual activation function. However, the computational burden is drastically reduced, which is especially significant when dealing with simple networks (less than six hidden nodes).

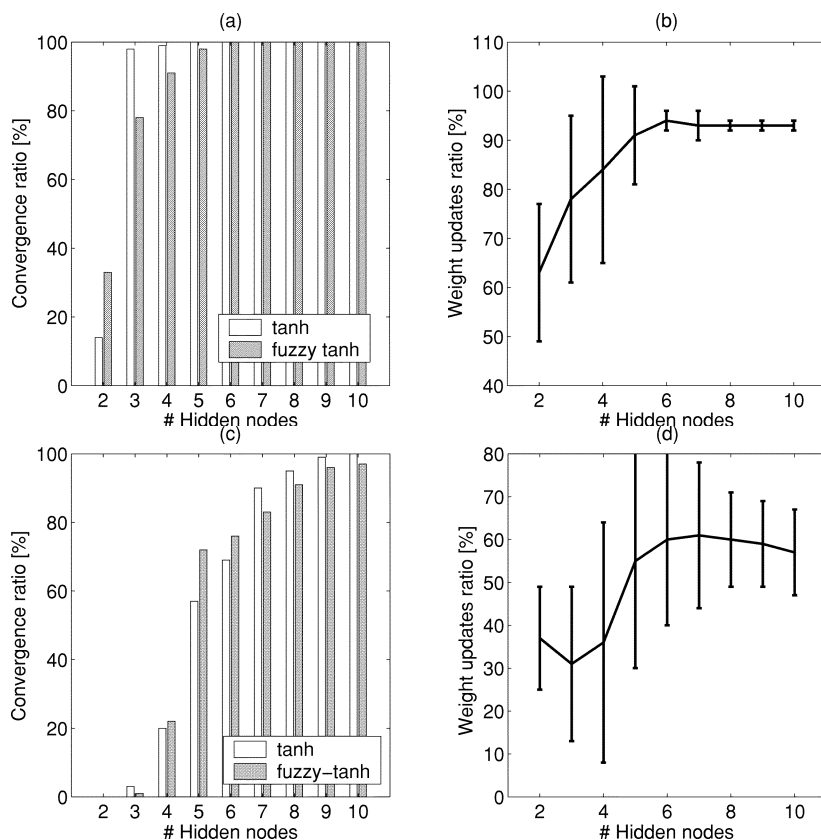


Fig. 3. Channel equalization using the standard hyperbolic tangent (\tanh) and the fuzzy tanh proposed. (a) Convergence ratio. (b) Error bar of the updating rate as a function of the number of hidden nodes considered for channel $H_1(z)$. (c) Convergence ratio. (d) Error bar of the updating rate as a function of the number of hidden nodes considered for channel $H_2(z)$.

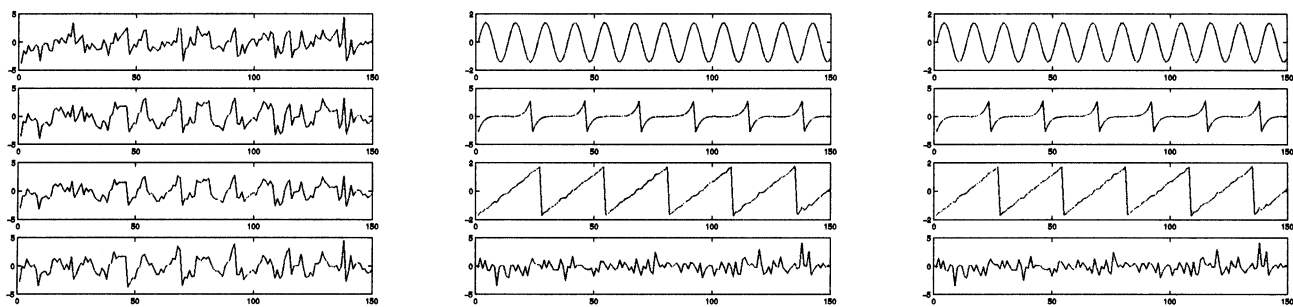


Fig. 4. Left: Original signals. Middle: Obtained signals with the \tanh in the fast ICA. Right: Obtained signals with the fuzzy tanh using the fast ICA.

D. Independent Component Analysis

The last experiment deals with the use of the fuzzy tanh in ICA [6] of mixed signals by using the fixed-point fast ICA implementation, which is available from <http://www.cis.hut.fi/projects/ica/fastica/>. We have used a mixture of four signals: sinusoid, sawtooth, impulsive noise, and “funny” signal, provided within the fast ICA toolbox. Results with both implementations are shown in Fig. 4. No numerical or statistically significant differences are appreciable between both implementations. However, the computational burden and simplicity involved in our proposal make it a better choice.

IV. RULE GENERATION AND INTERPRETABILITY

Rule generation from a trained network with the proposed activation function can be done by analyzing the saturated zones of its fuzzy ac-

tivation functions $f(\mathbf{x}) = \pm 1$. In fact, given the weight vector \mathbf{w}_k and bias s_k , the saturated zones of a neuron k can be defined as

$$\mathbf{w}_k \cdot \mathbf{x} + s_k \geq L \quad (7)$$

$$\mathbf{w}_k \cdot \mathbf{x} + s_k < -L. \quad (8)$$

The left-hand terms of inequalities are the parallel hyperplane equations, which are separated a distance $2L$, as illustrated in Fig. 5, in a bidimensional input space.

These zones allow rule extraction since they can be combined to produce a determined output function. Two saturated zones can be obtained for every output node. Therefore, the number of rules is 2^N , N being the number of hidden neurons. In the next example, we show the weights and induced rules from a trained network in the XOR-gate problem. The chosen architecture was $2 \times 2 \times 1$ for illustration purposes.

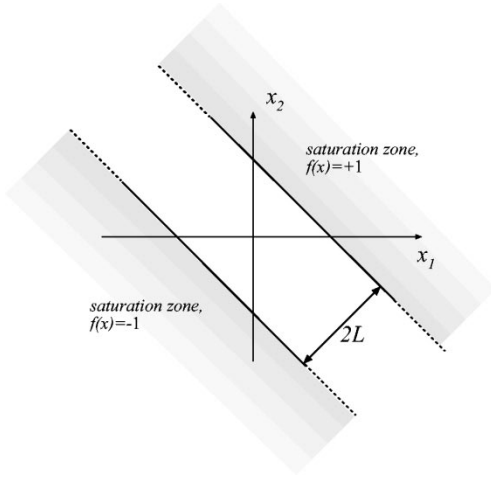


Fig. 5. Saturated zones in a bidimensional input space.

- Synaptic weights in input to hidden layer

$$\mathbf{w}_1 = \{2.06, 8.24\}, \quad s_1 = 7.60 \quad (9)$$

$$\mathbf{w}_2 = \{-2.51, 0.05\}, \quad s_2 = -13.08 \quad (10)$$

where saturation zones of (9) are

$$2.06x_1 + 8.24x_2 + 7.6 > 1$$

$$2.06x_1 + 8.24x_2 + 7.6 < -1$$

and the saturation zones of (10) are

$$-2.51x_1 + 0.05x_2 - 13.08 > 1$$

$$-2.51x_1 + 0.05x_2 - 13.08 < -1.$$

- Synaptic weights in hidden to output layer

$$\mathbf{w}_3 = \{-1.63, -5.04\}, \quad s_1 = -2.55 \quad (11)$$

where the saturation zones of (11) are

$$-1.63u_1 - 5.04u_2 - 2.55 > 1$$

$$-1.63u_1 - 5.04u_2 - 2.55 < -1$$

and u_1, u_2 represent the hidden outputs.

Since two hidden nodes are present, only four rules are generated. Only one is shown here to illustrate the general procedure. For instance,

if $u_1 = 1$ and $u_2 = 1$, the output is $y = -1$. This rule can be extracted easily from the saturation zones by

$$\begin{aligned} \mathcal{R}_1 : & \text{ If } 2.06x_1 + 8.24x_2 > -6.6 \\ & \text{ and } -2.51x_1 + 0.05x_2 > 14.08 \\ & \text{ then } y = -1. \end{aligned} \quad (12)$$

Therefore, rule \mathcal{R}_1 can be established as the superposition of two hyperplanes in the input space, which can be defined by the relationship of other input variables easily, as

$$\begin{aligned} \mathcal{R}_2 : & \text{ If } x_2 > 0.25x_1 - 0.8 \\ & \text{ and } x_2 > 50x_1 + 281.6 \\ & \text{ then } y = -1. \end{aligned} \quad (13)$$

Nevertheless, we must remark that in the nonsaturated zones, rule extraction would not be an immediate task. However, saturated zones cover most of the searching space, which is undoubtedly an interesting property for rule extraction. We finally conclude that the use of this activation function enables a straightforward implementation of a neural network and facilitates rule generation from a trained model.

V. CONCLUSION

We have proposed the use of concepts from fuzzy logic to develop an activation function similar to those commonly used in ANNs: the hyperbolic tangent and the sigmoid. The new function can be implemented in a simple way in a hardware device. On the other hand, the proposed approach allows the development of activation functions using other membership functions. Additionally, a modification of the classical backpropagation learning algorithm is proposed. Typical problems in ANNs were simulated in order to show the validity of the approach. In addition, the ability to extract rules based on the combination of hyperplanes in the input space is shown. Further work will consider the use of other membership functions such as Gaussian or bell-shaped.

REFERENCES

- [1] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice-Hall, 1999.
- [2] H. K. Kwan, "Simple sigmoid-like activation function suitable for digital hardware implementation," *Inst. Elect. Eng. Electron. Lett.*, vol. 28, pp. 1379–1380, 1992.
- [3] G. J. Klir, U. H. Clair, and B. Yuan, *Fuzzy Set Theory: Foundations and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1997.
- [4] M. Casdagli and S. Eubank, "Nonlinear modeling and forecasting," in *Proceedings of the Santa Fe Institute Studies in the Science of Complexity*. Reading, MA: Addison-Wesley, 1992, vol. XII.
- [5] G. J. Gibson, S. Siu, and C. F. N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals," *IEEE Trans. Signal Processing*, vol. 39, pp. 1877–1884, 1991.
- [6] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.