

Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks

Bekir Karlik

*Engineering Faculty/Computer Engineering/
Mevlana University, Konya, 42003,
Turkey*

bkarlik@mevlana.edu.tr

A Vehbi Olgac

*Engineering Faculty/Computer Engineering/
Fatih University, Istanbul, 34500,
Turkey*

avehbi@live.com

Abstract

The activation function used to transform the activation level of a unit (neuron) into an output signal. There are a number of common activation functions in use with artificial neural networks (ANN). The most common choice of activation functions for multi layered perceptron (MLP) is used as transfer functions in research and engineering. Among the reasons for this popularity are its boundedness in the unit interval, the function's and its derivative's fast computability, and a number of amenable mathematical properties in the realm of approximation theory. However, considering the huge variety of problem domains MLP is applied in, it is intriguing to suspect that specific problems call for single or a set of specific activation functions. The aim of this study is to analyze the performance of generalized MLP architectures which has back-propagation algorithm using various different activation functions for the neurons of hidden and output layers. For experimental comparisons, Bi-polar sigmoid, Uni-polar sigmoid, Tanh, Conic Section, and Radial Bases Function (RBF) were used.

Keywords: Activation Functions, Multi Layered Perceptron, Neural Networks, Performance Analysis

1. INTRODUCTION

One of the most attractive properties of ANNs is the possibility to adapt their behavior to the changing characteristics of the modeled system. Last decades, many researchers have investigated a variety of methods to improve ANN performance by optimizing training methods, learn parameters, or network structure, comparably few works is done towards using activation functions. Radial basis function (RBF) neural network is one of the most popular neural network architectures [1]. The standard sigmoid reaches an approximation power comparable to or better than classes of more established functions investigated in the approximation theory (i.e., splines and polynomials) [2]. Jordan presented the logistic function which is a natural representation of the posterior probability in a binary classification problem [3]. Liu and Yao improved the structure of Generalized Neural Networks (GNN) with two different activation function types which are sigmoid and Gaussian basis functions [4]. Sopena et al. presented a number of experiments (with widely-used benchmark problems) showing that multilayer feed-forward networks with a sine activation function learn two orders of magnitude faster while generalization capacity increases (compared to ANNs with logistic activation function) [5]. Dorffner developed the Conic Section Function Neural Network (CSFNN) which is a unified framework for MLP and RBF networks to make simultaneous use of advantages of both networks [6]. Bodyanskiy presented a novel double-wavelet neuron architecture obtained by modification of standard wavelet neuron, and their learning algorithms are proposed. The proposed architecture allows improving the

approximation properties of wavelet neuron [7]. All these well-known activations functions are used into nodes of each layer of MLP to solve different non-linear problems. But, there is no performance of comparison studies using these activation functions. So, in this study, we have used five different well-known activation functions such as Bi-polar sigmoid, Uni-polar sigmoid, Tanh, Conic Section, and Radial Bases Function (RBF) to compare their performances.

2. ACTIVATION FUNCTION TYPES

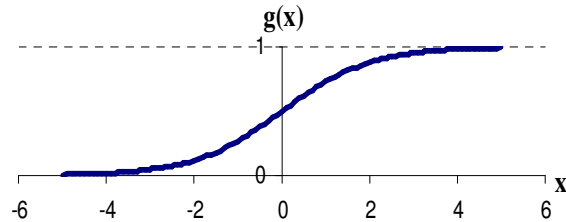
The most important unit in neural network structure is their net inputs by using a scalar-to-scalar function called “the activation function or threshold function or transfer function”, output a result value called the “unit's activation”. An activation function for limiting the amplitude of the output of a neuron. Enabling in a limited range of functions is usually called squashing functions [8-9]. It squashes the permissible amplitude range of the output signal to some finite value. Some of the most commonly used activation functions are to solve non-linear problems. These functions are: Uni-polar sigmoid, Bi-polar sigmoid, Tanh, Conic Section, and Radial Bases Function (RBF). We did not care about some activation function such as identity function, step function or binary step functions as they are not used to solve linear problems.

2.1 Uni-Polar Sigmoid Function

Activation function of Uni-polar sigmoid function is given as follows:

$$g(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

This function is especially advantageous to use in neural networks trained by back-propagation algorithms. Because it is easy to distinguish, and this can interestingly minimize the computation capacity for training. The term sigmoid means ‘S-shaped’, and logistic form of the sigmoid maps



the interval $(-\infty, \infty)$ onto $(0, 1)$ as seen in **FIGURE 1**.

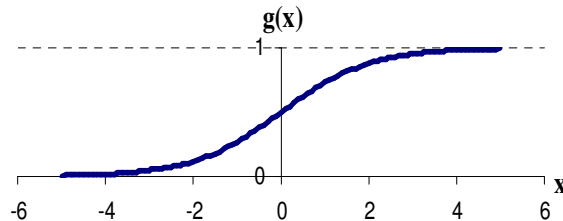


FIGURE 1: Uni-Polar Sigmoid Function

2.2 Bipolar Sigmoid Function

Activation function of Bi-polar sigmoid function is given by

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2)$$

This function is similar to the *sigmoid function*. For this type of activation function described in Fig. 2, it goes well for applications that produce output values in the range of $[-1, 1]$.

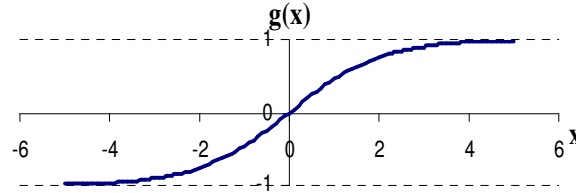


FIGURE 2: Bi-Polar Sigmoid Function

2.3 Hyperbolic Tangent Function

This function is easily defined as the ratio between the hyperbolic sine and the cosine functions or expanded as the ratio of the half-difference and half-sum of two exponential functions in the points x and $-x$ as follows :

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

Hyperbolic Tangent Function is similar to sigmoid function. Its range outputs between -1 and 1 as seen in

FIGURE . The following is a graphic of the hyperbolic tangent function for real values of its argument x :

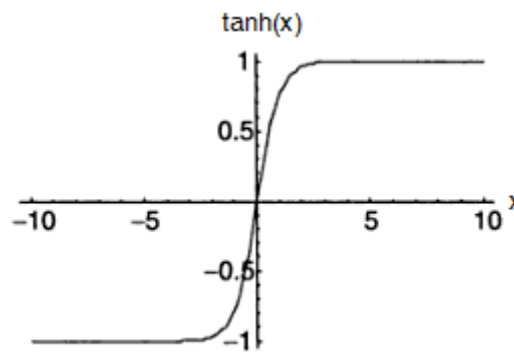


FIGURE 3: Hyperbolic Tangent Function

2.4 Radial Basis Function

Radial basis function (RBF) is based on Gaussian Curve. It takes a parameter that determines the center (mean) value of the function used as a desired value (see Fig. 4). A radial basis

function (RBF) is a real-valued function whose value depends only on the distance from the origin, so that

$$g(x) = g(\|x\|) \quad (4)$$

or alternatively on the distance from some other point c , called a *center*, so that

$$g(x, c) = g(\|x - c\|) \quad (5)$$

Sums of radial basis functions are typically used to approximate given functions. This approximation process can also be interpreted as a simple type of neural network.

RBF are typically used to build up function approximations of the form

$$y(x) = \sum_{i=1}^N w_i g(\|x - c_i\|) \quad (6)$$

where the approximating function $y(x)$ is represented as a sum of N radial basis functions, each associated with a different center c_i , and weighted by an appropriate coefficient w_i . The weights w_i can be estimated using the matrix methods of linear least squares, because the approximating function is *linear* in the weights. Fig. 4 shows that two unnormalized Gaussian radial basis functions in one input dimension. The basis function centers are located at $c_1=0.75$ and $c_2=3.25$ [10].

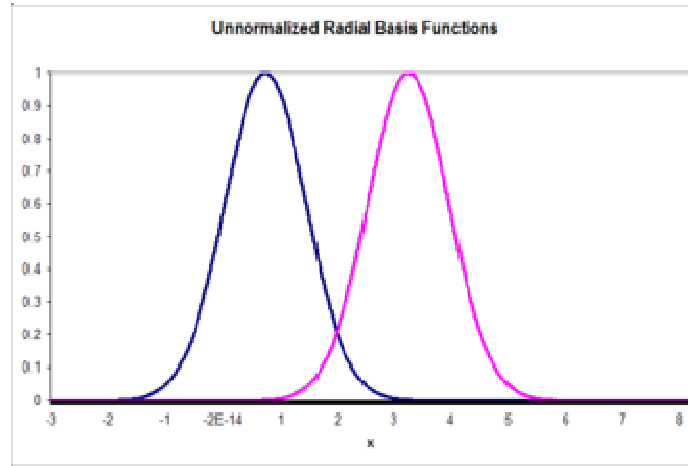


FIGURE 4: Two unnormalized Gaussian radial basis functions in one input dimension

RBF can also be interpreted as a rather simple single-layer type of artificial neural network called a radial basis function network, with the radial basis functions taking on the role of the activation functions of the network. It can be shown that any continuous function on a compact interval can in principle be interpolated with arbitrary accuracy by a sum of this form, if a sufficiently large number N of radial basis functions is used.

2.5 Conic Section Function

Conic section function (CSF) is based on a section of a cone as the name implies. CSF takes a parameter that determines the angle value of the function as seen in Fig. 5.

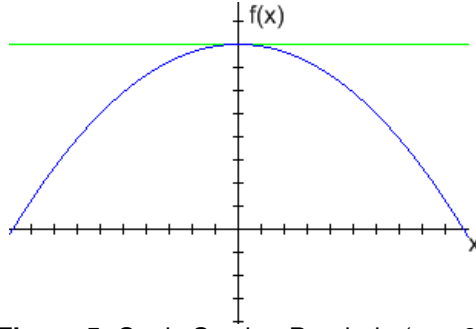


Figure 5: Conic Section Parabola ($\omega = 90$)

The equation of CSF can be defined as follows:

$$f(x) = \sum_{i=1}^{N+1} (a_i - c_i) w_i - \cos w_i (\|a - c_i\|) \quad (7)$$

where a_i is input coefficient, c_i is the center, w_i is the weight in Multi Layered Perceptron (MLP), $2w$ is an opening angle which can be any value in the range of $[-\pi/2, \pi/2]$ and determines the different forms of the decision borders.

The hidden units of neural network need activation functions to introduce non-linearity into the networks. Non-linearity makes multi-layer networks so effective. The sigmoid functions are the most widely used functions [10-11]. Activation functions should be chosen to be suited to the distribution of the target values for the output units. You can think the same for binary outputs where the tangent hyperbolic and sigmoid functions are effective choices. If the target values are positive but have no known upper bound, an exponential output activation function can be used. This work has explained all the variations on the parallel distributed processing idea of neural networks. The structure of each neural network has very similar parts which perform the processing.

3. COMPARISON WITH DIFFERENT ACTIVATION FUNCTIONS

In this study, different activation functions depend on different number of iterations for comparing their performances by using the same data. For all the activation functions, we used the number of nodes in the hidden layer; firstly 10 nodes, secondly 40 nodes (numbers of iterations are the same for both of them). After presenting the graphs for different parameters from Figure 6 through Figure 10, interpretations of their results will follow right here.

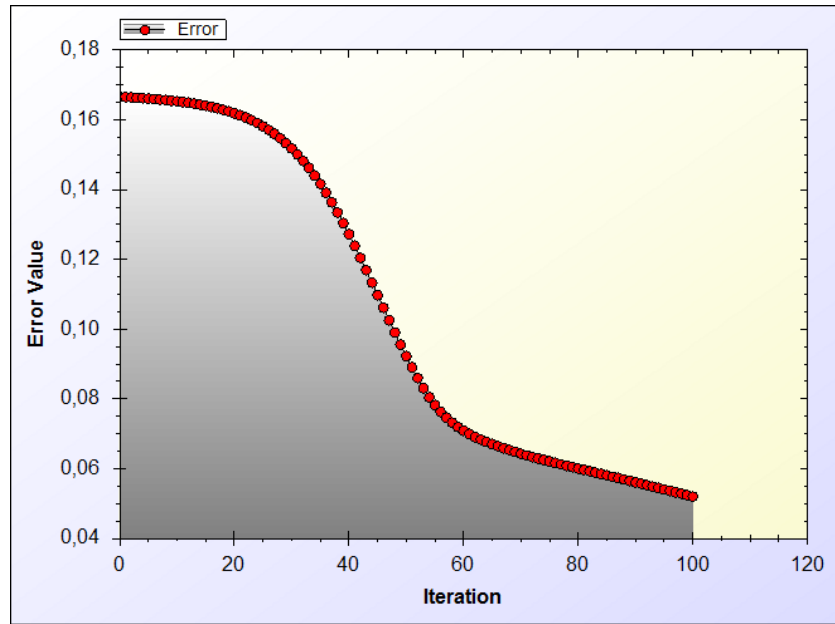


FIGURE 6: 100 Iterations - 10 Hidden Neurons - Bi-Polar Sigmoid

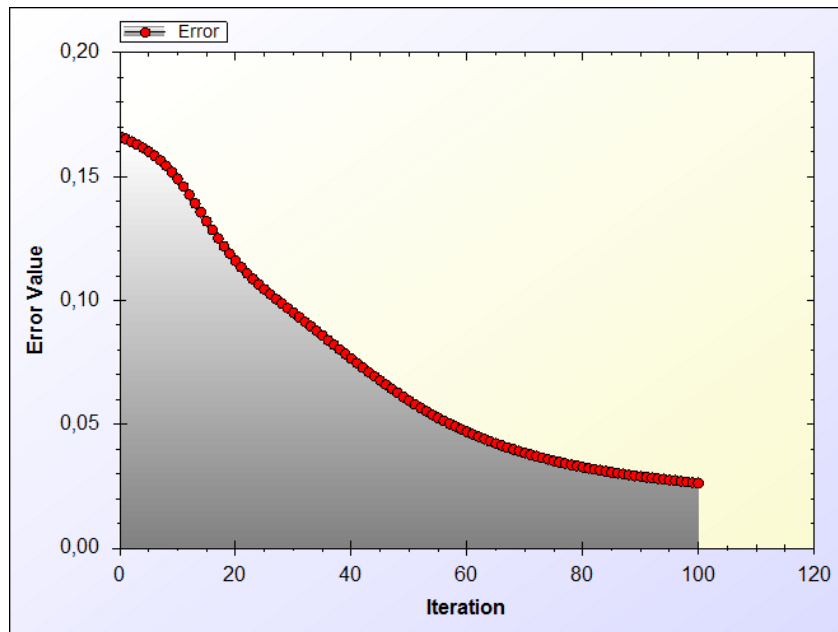


FIGURE 7: 100 Iterations - 10 Hidden Neurons - Uni-Polar Sigmoid

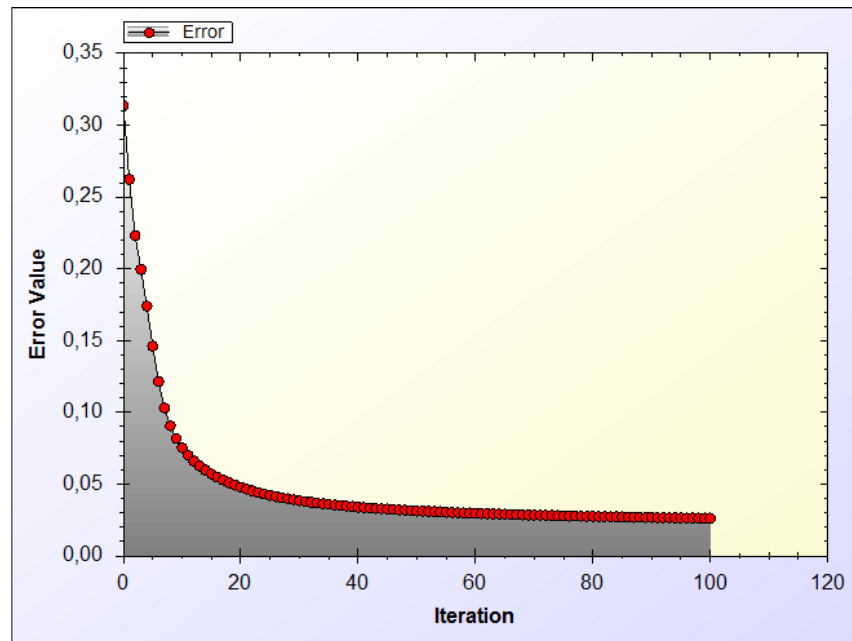


FIGURE 8: 100 Iterations - 10 Hidden Neurons - Tangent Hyperbolic

We used the same number of hidden neurons (10 nodes) and iterations (100 iterations) to compare the differences between the activation functions used above. It is also requires less number of iterations.

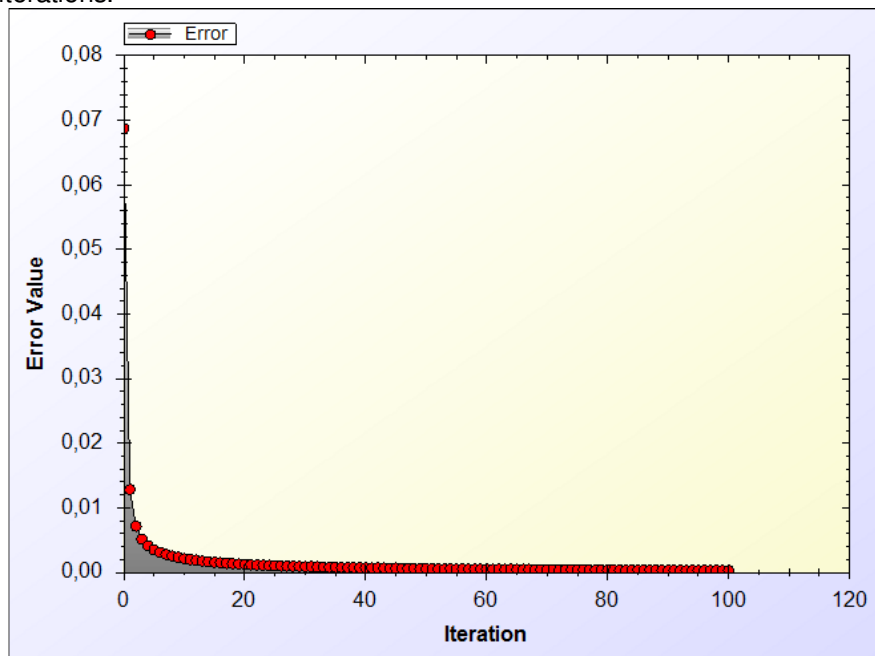


FIGURE 9: 100 Iterations - 10 Hidden Neurons - Conic Section

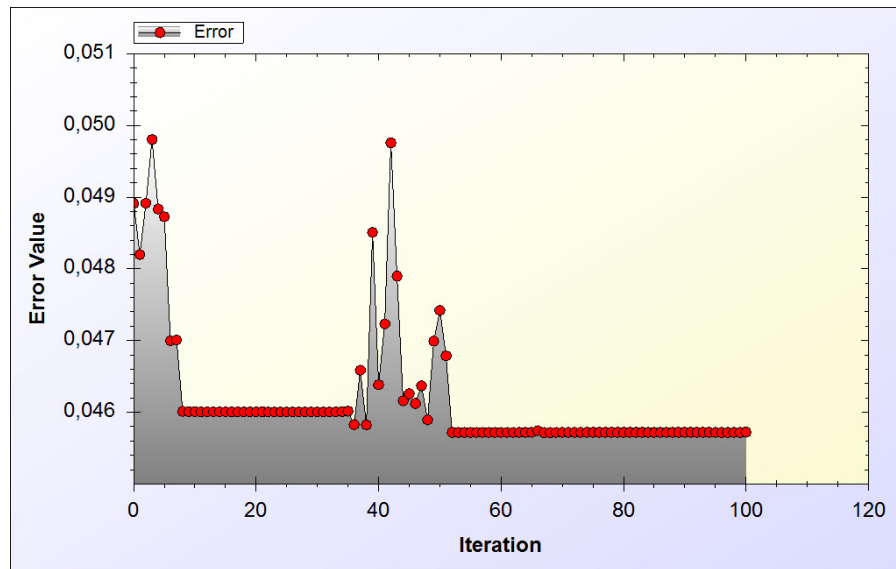


FIGURE 10: 100 Iterations - 10 Hidden Neurons – RBF

According to the Figures from 6 through 15, we found Conic Section function the best activation function for training. Moreover, it requires less number of iterations than the other activation functions to solve the non-linear problems. However, with regard to testing, we found that the accuracy of Tanh activation function was much better than CSF and the other activation functions. This situation explains that total Mean Square Error (MSE) according to iterations cannot determine the network accuracy. Hence, we can obtain the real accuracy results upon testing.

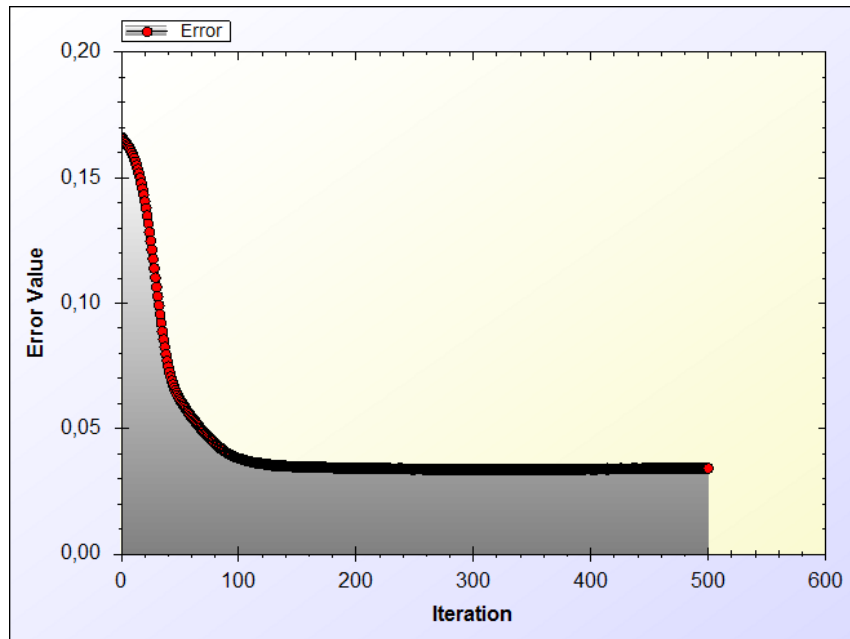


FIGURE 11: 500 Iterations - 40 Hidden Neurons - Bi-Polar Sigmoid

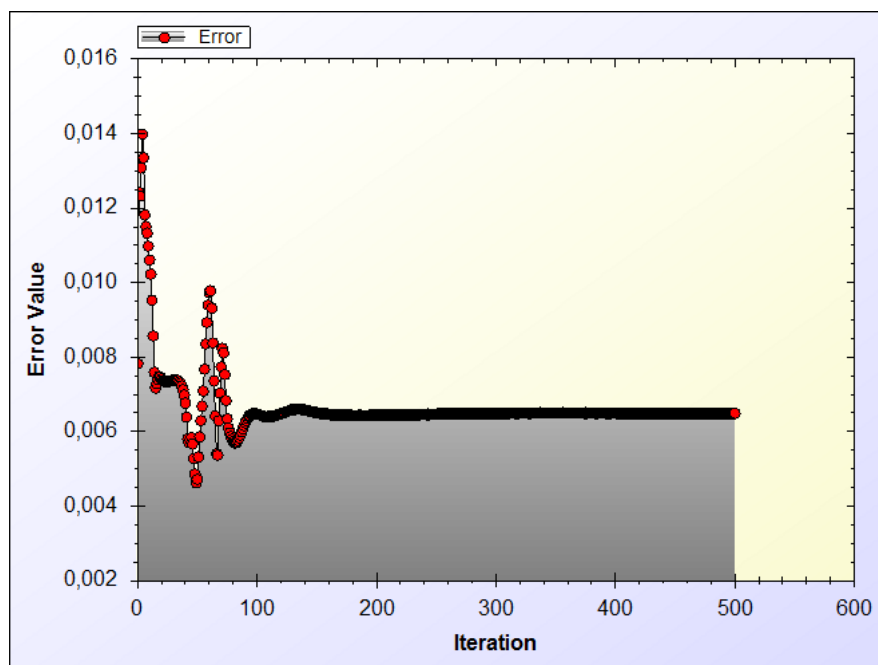


FIGURE 12: 500 Iterations - 40 Hidden Neurons - Uni-Polar Sigmoid

Figures 11 through 15 show the error graphics for each activation function for different numbers of nodes in the hidden layer which may vary from 10 up to 40, and the number of iterations from 100 up to 500.

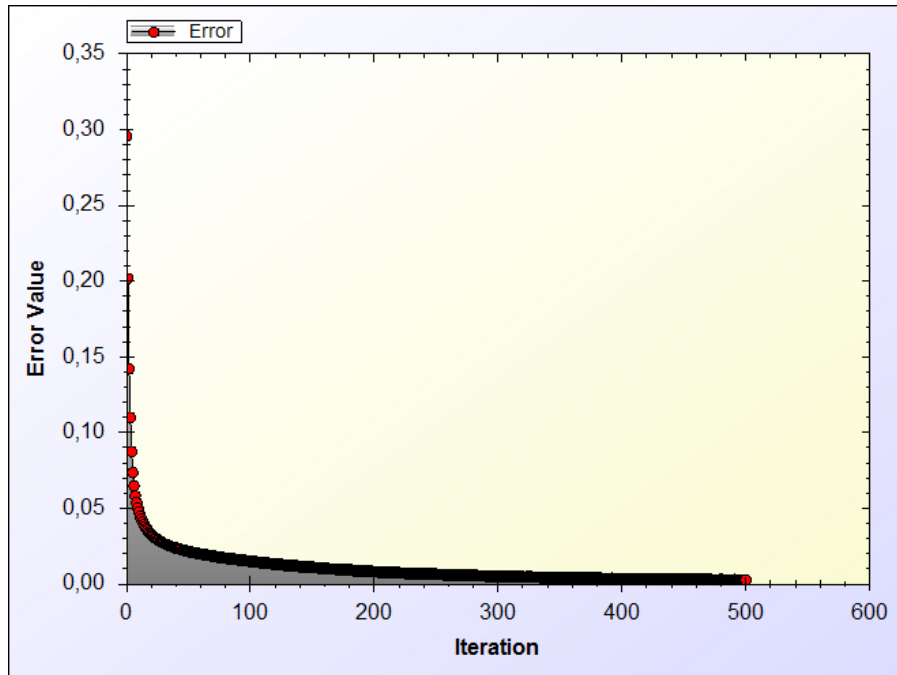


FIGURE 13: 500 Iterations - 40 Hidden Neurons - Tangent Hyperbolic

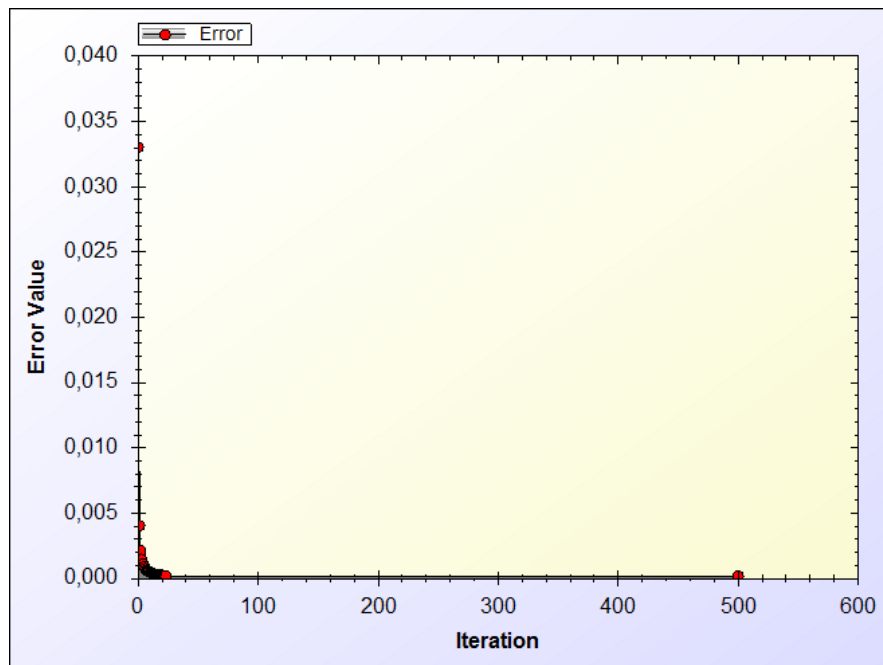


FIGURE 14: 500 Iterations - 40 Hidden Neurons - Conic Section

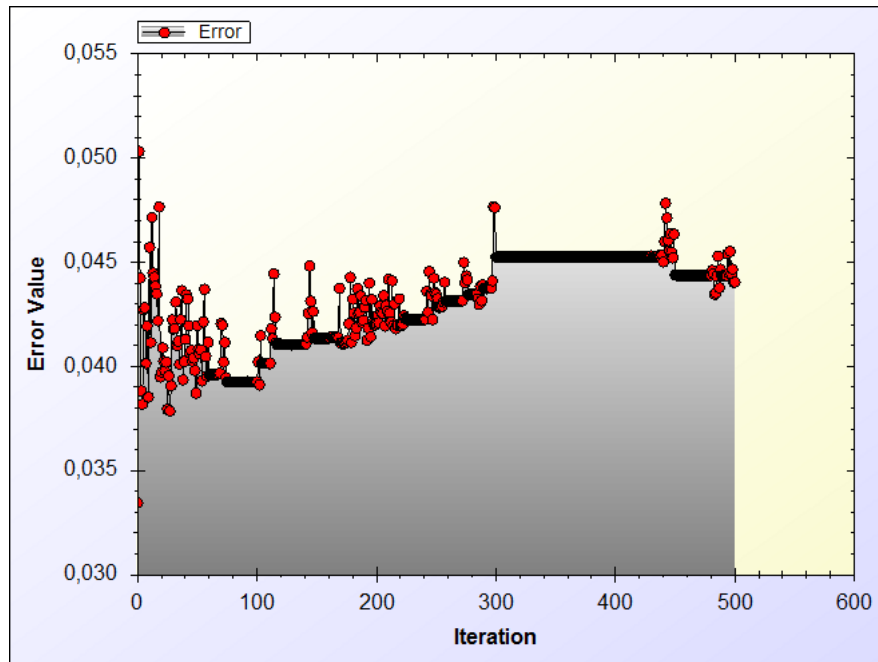


FIGURE 15: 500 Iterations - 40 Hidden Neurons – RBF

According to the first five graphics generated by the same iteration numbers and the same number of neurons in the hidden layer, the Tangent Hyperbolic Activation function resulted in the most successful one for all the test parameters. Conic Section Function and Radial Basis Function were not able to successfully use the data type and the network parameters. The second group of result graphs is similar to the first group. But, the accuracy of the algorithms was observed to increase for the sigmoid and tangent hyperbolic functions. The rest of the functions used in this study were not successful and not accurate enough in this group. Table 1 shows those total accuracy and error values for the testing phase.

	100 Iterations 10 Hidden Neurons		500 Iterations 40 Hidden Neurons	
	Error	Accuracy (%)	Error	Accuracy (%)
Bi-Polar S.	0,056	93	0,034	89
Uni-Polar S.	0,026	92	0,006	88
Tanh	0,025	95	0,002	99
Conic	0,001	34	0,045	23
RBF	0,003	30	0,001	19

TABLE 1: Results of the testing phase

4. CONCLUSIONS

In this study, we have used five conventional differentiable and monotonic activation functions for the evolution of MLP architecture along with Generalized Delta rule learning. These proposed well-known and effective activation functions are Bi-polar sigmoid, Uni-polar sigmoid, Tanh, Conic Section, and Radial Bases Function (RBF). Having compared their performances, simulation results show that Tanh (hyperbolic tangent) function performs better recognition accuracy than those of the other functions. In other words, the neural network computed good results when "Tanh-Tanh" combination of activation functions was used for both neurons (or nodes) of hidden and output layers.

We presented experimental results by proposing five different activation functions for MLP neural networks architecture along with Generalized Delta Rule Learning to solve the non-linear problems. These results demonstrate that it is possible to improve the ANN performance through the use of much effective activation function. According to our experimental study, we can say that the Tanh activation function can be used in the vast majority of MLP applications as a good choice to obtain high accuracy. Furthermore, the sole use of validation test cannot approve the work even if it would be able to predict very low MSE. So, we can only decide about the real accuracy after a thorough testing of the neural network. In the future work, non-monotonic activation functions will probably be sought for the test results in the learning speed and the performance of the neural network.

REFERENCES

1. T. Poggio, F. Girosi, "A theory of networks for approximation and learning". A.I. Memo No. 1140, Artificial Intelligence, Laboratory, Massachusetts Institute of Technology, 1989
2. B. DasGupta, G. Schnitger, "The Power of Approximating: A Comparison of Activation Functions". In Giles, C. L., Hanson, S. J., and Cowan, J. D., editors, *Advances in Neural Information Processing Systems*, 5, pp. 615-622, San Mateo, CA. Morgan Kaufmann Publishers, 1993
3. M.I. Jordan, "Why the logistic function? A tutorial discussion on probabilities and neural networks". Computational Cognitive Science Technical Report 9503, Massachusetts Institute of Technology, 1995
4. Y. Liu, X. Yao, "Evolutionary Design of Artificial Neural Networks with Different Nodes". In Proceedings of the Third IEEE International Conference on Evolutionary Computation, pp. 570-675, 1996
5. J. M. Sopena, E. Romero, R. Alqu  azar, "Neural networks with periodic and monotonic activation functions: a comparative study in classification problems". In Proceedings of the 9th International Conference on Artificial Neural Networks, pp. 323-328, 1999
6. G. Dorffner, "Unified frameworks for MLP and RBFNs: Introducing Conic Section Function Networks". *Cybernetics and Systems*, 25: 511-554, 1994
7. Y. Bodyanskiy, N. Lamonova, O. Vynokurova, "Double-Wavelet Neuron Based on Analytical Activation Functions". *International Journal Information Theories & Applications*, 14: 281-288, 2007
8. G. Cybenko, "Approximation by superposition of a sigmoidal function". *Mathematics of Control, Signals, and Systems*, 2(4):303-314, 1987
9. R. P. Lippmann, "An introduction to computing with neural nets". *IEEE Acoustics, Speech and Signal Processing*, 4(2):4-22, 1987
10. M.D. Buhmann, (2003), *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, ISBN 978-0-521-63338-3.
11. B. Widrow, M.A. Lehr, "30 years of adoptive neural networks; perceptron, madaline, and back propagation". *Proc. IEEE*, 78: 1415-1442, 1990
12. B. Ciocoiu, "Hybrid Feedforward Neural Networks for Solving Classification Problems". *Neural Processing Letters*, 16(1):81-91, 2002