XVI IMEKO World Congress
Measurement - Supports Science - Improves Technology - Protects Environment ... and Provides Employment - Now and in the Future
Vienna, AUSTRIA, 2000, September 25-28

ESTIMATING THE ACTIVATION FUNCTIONS OF AN MLP-NETWORK

P.V. Vehviläinen, H.A.T. Ihalainen
Laboratory of Measurement and Information Technology
Automation Department
Tampere University of Technology, FIN-33101, Tampere, Finland

Abstract: The most common neural network type is a multilayer perceptron (MLP) network [1]. Even a relatively simple MLP-network requires usually high computational accuracy and efficiency. The usual types of activation functions are not necessarily applicable for a neural network algorithm used in microcontrollers. For this purpose a new type of activation function is introduced.

Keywords: Neural network, Activation Function, Estimation, Multilayer Perceptron

1 INTRODUCTION

An MLP-network (Multilayer Perceptron) typically consists of a set of sensory units (an input layer), one or more hidden layers and an output layer. The hidden layer(s) and the output layer are actually a set of computational nodes. These nodes or perceptrons sum the signals attached to it and employ an activation function to this sum. Figure 1 illustrates this type of a perceptron. The product of every input signal, x_k and its corresponding weight parameter w_{kp} are summed in an adder Σ. This sum v_k is then given as an argument to an activation function $\varphi(\cdot)$. The output of this function is also the output of the perceptron.

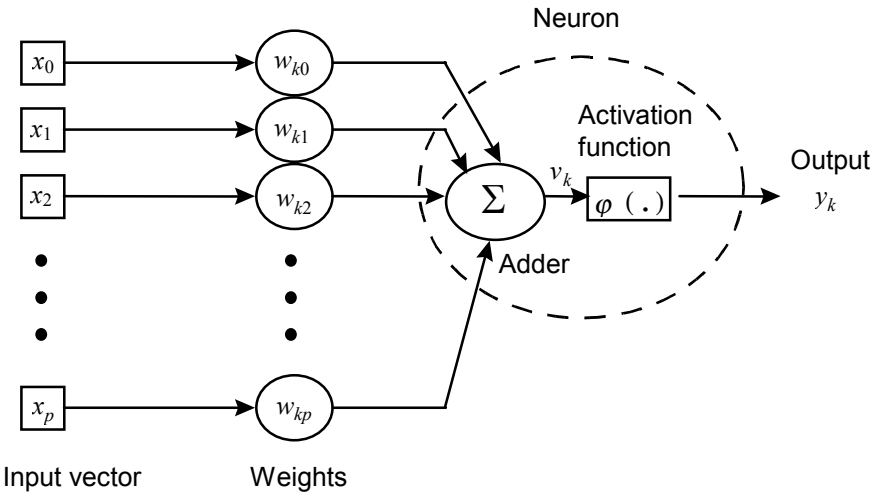

Figure 1. Signal flow chart for a perceptron.

The most important benefit of an MLP-network is its ability to perform a non-linear input-output mapping [2]. To achieve this mapping the basic units i.e. the activation functions have to be non-linear. Furthermore, it is required from the activation functions that they must be continuous and differentiable as well. Two very popular examples of such activation functions are the *Logistic Function* (also often referred as the *Logistic Sigmoid*) represented in Eq. (1) and the *Hyperbolic Tangent Function* in Eq. (2). These functions are shown with their derivatives in Figures 2 and 3.

$$\varphi_{\log}(v_k) = \frac{1}{1+\exp(-v_k)} \tag{1}$$

$$\varphi_{\tanh}(v_k) = \tanh(v_k) = \frac{e^{v_k} - e^{-v_k}}{e^{v_k} + e^{-v_k}} = \frac{1 - \exp(-2v_k)}{1 + \exp(-2v_k)} \tag{2}$$

It is easy to see that the hyperbolic tangent is merely the logistic function biased and rescaled: by using constants a and b:

$$a \tanh(b\, v_k) = a\left[\frac{1 - \exp(-2\, b\, v_k)}{1 + \exp(-2\, b\, v_k)}\right] = \frac{2\, a}{1 + \exp(-2\, b\, v_k)} - a \tag{3}$$

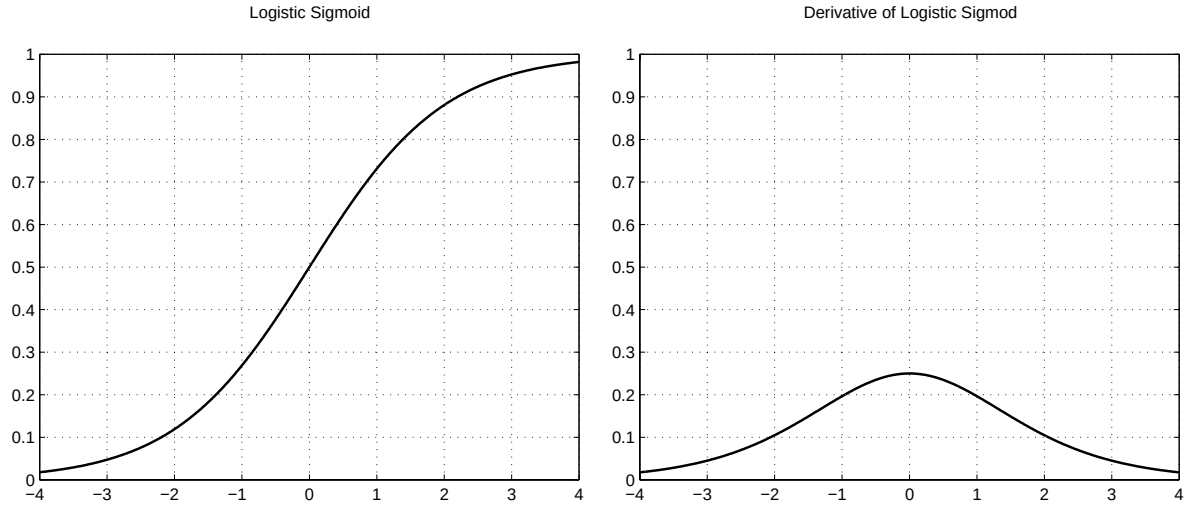

Figure 2. Logistic sigmoid and its derivative.

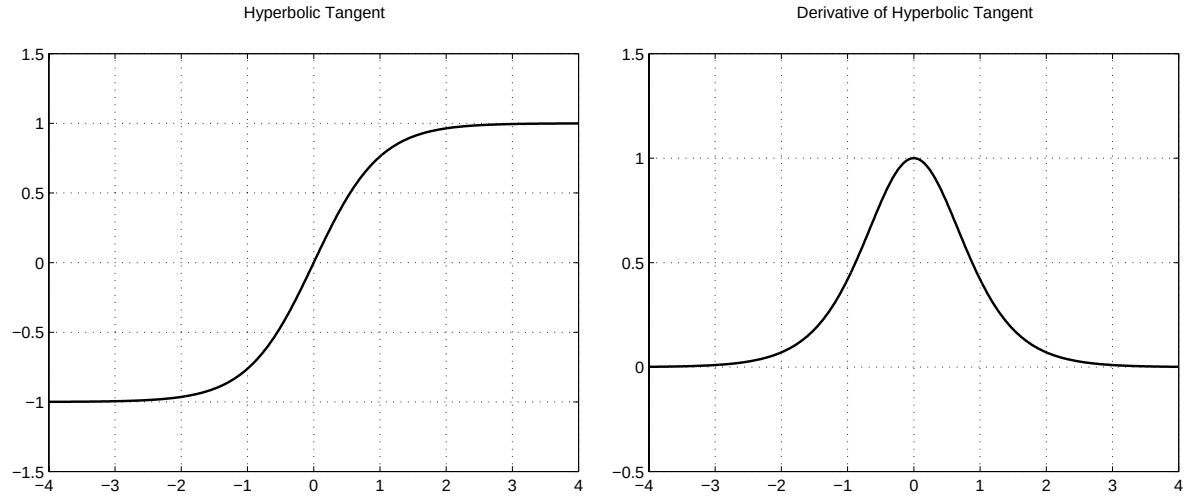

Figure 3. Hyperbolic tangent and its derivative.

The derivatives of these functions are expressed in Eqs. (4) and (5) for logistic function and hyperbolic tangent, respectively:

$$\varphi'_{\log}(v_k) = \frac{1}{1 + \exp(-v_k)}\left(1 - \frac{1}{1 + \exp(-v_k)}\right) \tag{4}$$

$$\varphi'_{\tanh}(v_k) = \frac{1}{\cosh^2(v_k)} = \left(1 - \frac{e^{v_k} - e^{-v_k}}{e^{v_k} + e^{-v_k}}\right)\left(1 + \frac{e^{v_k} - e^{-v_k}}{e^{v_k} + e^{-v_k}}\right) \tag{5}$$

2 PROPOSED ACTIVATION FUNCTIONS

It can be seen from the previous functions that they (and their derivatives) require the use of the exponential function. The exponential function is understood to have Napier's constant $e \approx 2{,}7182818285...$ as the mantissa and the input argument as the exponent. When using the exponential function in neural networks, the input argument for the exponential function is always a floating-point figure. For simulations in computer workstations this poses no problem, as the exponent function is allowed and the computational accuracy is commonly 64-bit. However, when using computationally simpler microcontrollers it is possible that an operation needing floating-point figure powered by another one is not allowed.

Should this be the case, computationally simpler activation functions are needed. For an activation function two things are required: first the function must be non-linear, and second, it must be continuously differentiable.

At first - trying to approximate the activation functions - an index table for the functions was created. Function values for corresponding input arguments were mapped into an index table. This approach was proven unsuccessful as the quantisation errors affected the learning process thus resulting in a network with unacceptable performance.

Second, a function approximations based on spline function approach was tried. Spline-based approach resulted in better network performance but was still not quite acceptable. Furthermore, the increased complexity and number of spline-functions compared to the original MLP-network were against their use in a microcontroller.

Finally, after these initial tests it was found out that it is possible to construct activation functions which do not need floating point powered by floating point calculation, and yet have close resemblance to the original activation functions. Such functions are asymptotic and even continuously differentiable. The function suggested to replace the logistic sigmoid function in Eq. (1) is represented in equation (6) and the function that is supposed to replace the hyperbolic tangent function in Eq. (2), is given in equation (7). It is shown in this paper that these functions can be justifiably used in replacing the original functions, which use the exponential function.

$$y_k = \begin{cases} 1 - \dfrac{\mathrm{a}}{1 + \left(0{,}5\, v_k + \mathrm{b}\right)^3}, & v_k \geq 0 \\ \dfrac{\mathrm{a}}{1 - \left(0{,}5\, v_k - \mathrm{b}\right)^3} - 1, & v_k < 0 \end{cases} \qquad \mathrm{a} = \frac{3}{4}; \quad \mathrm{b} = \frac{1}{\sqrt[3]{2}} \tag{6}$$

$$y_k = \begin{cases} 1 - \dfrac{\mathrm{a}}{1 + \left(v_k + \mathrm{b}\right)^3}, & v_k \geq 0 \\ \dfrac{\mathrm{a}}{1 - \left(v_k - \mathrm{b}\right)^3} - 1, & v_k < 0 \end{cases} \qquad \mathrm{a} = \frac{3}{2}; \quad \mathrm{b} = \frac{1}{\sqrt[3]{2}} \tag{7}$$

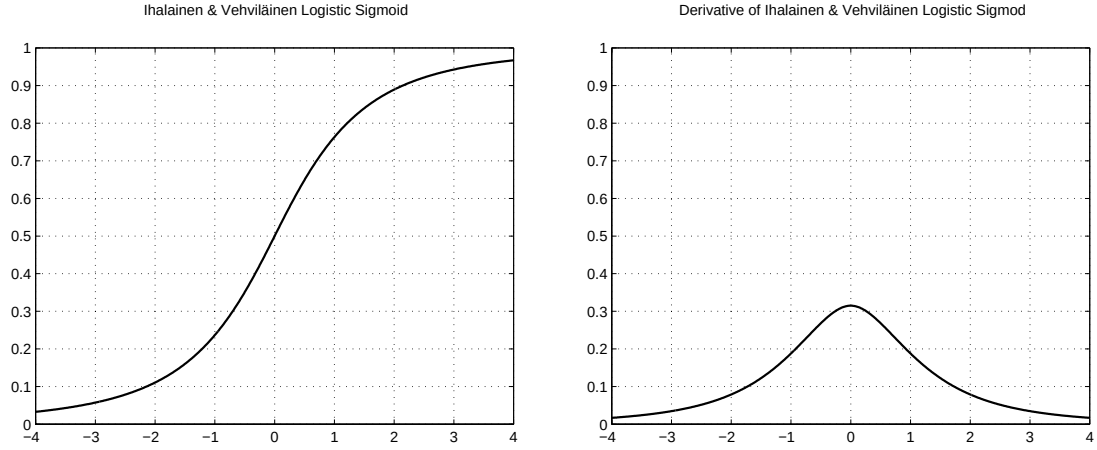

Figure 4. Proposed activation function (and its derivative) for replacing the logistic sigmoid.

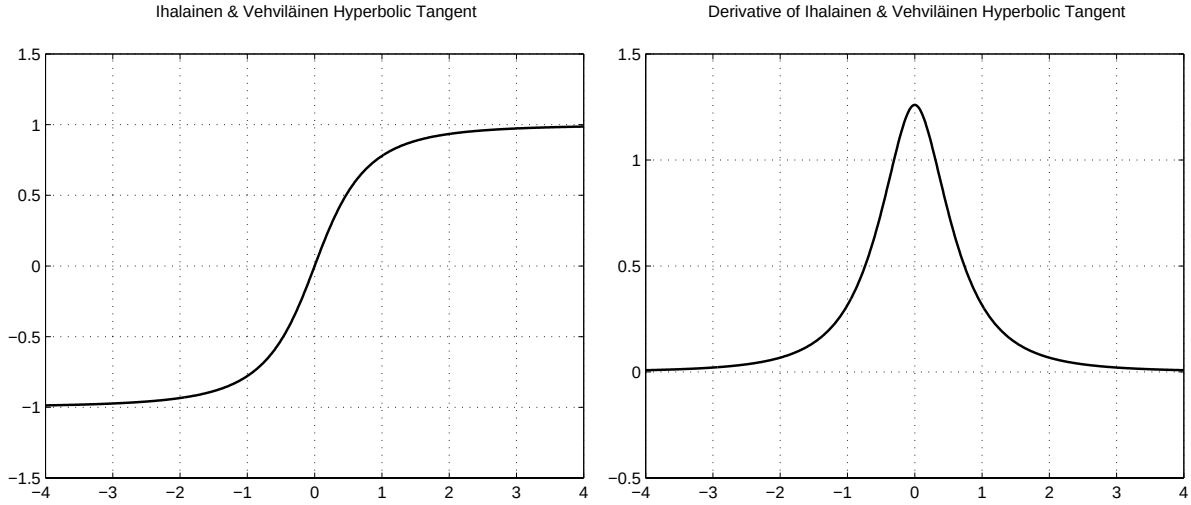

Figure 5. Proposed activation function (and its derivative) for replacing the hyperbolic tangent.

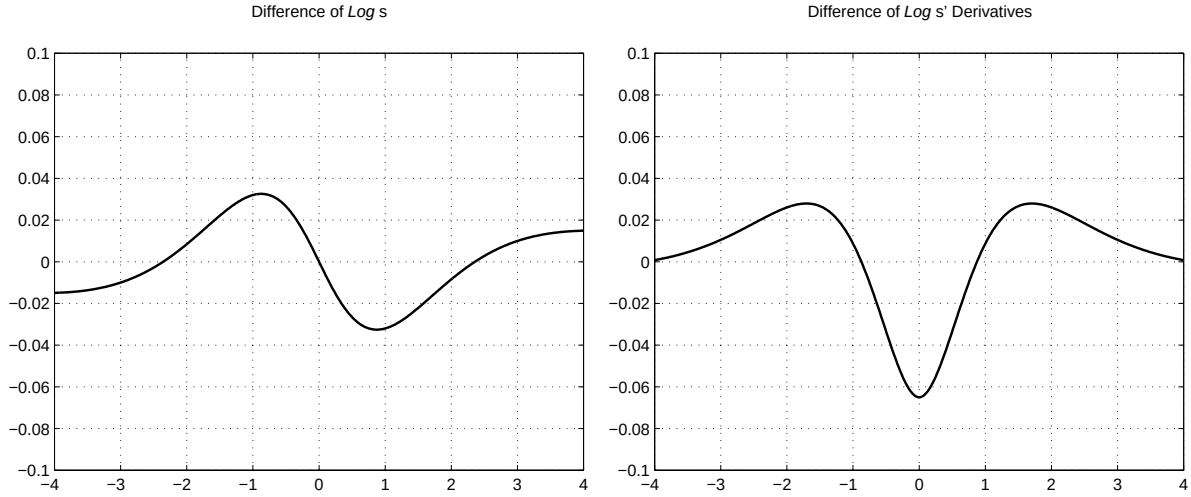

Figure 6. Difference of the original and proposed logistic sigmoids and their derivatives.

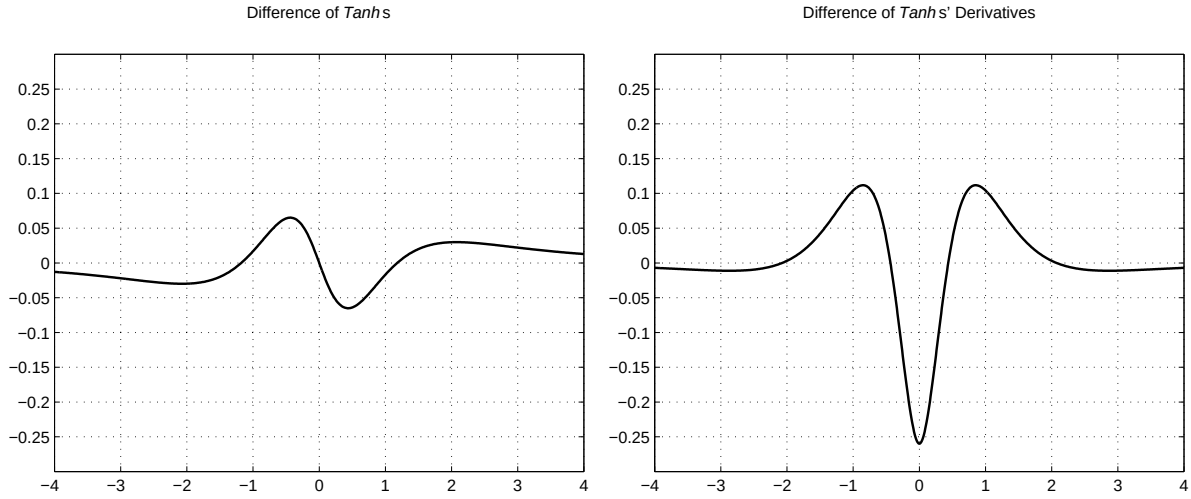

Figure 7. Difference of the original and proposed hyperbolic tangents and their derivatives.

3 TEST OF THE PROPOSED ACTIVATION FUNCTIONS

The proposed activation functions Eqs. (6) and (7) were compared with the original types of activation functions, Eqs. (1) and (2). These activation functions were tested in the following configuration: the data test set had 6 input vectors and one output vector, the MLP-network one hidden layer with 4 nodes, and an output layer with one node. For the original configuration hyperbolic tangents Eq. (2) were used in the hidden layer and logistic sigmoid Eq. (1) in the output layer. For the proposed network Eqs. (7) and (6) were used in the hidden and output layers, respectively. The training was conducted by using the standard back-propagation algorithm for 10 000 training epochs.

For both configurations 20 of such trainings were done. Both the training and the test data sets were real measurements acquired from an edge position measurement device. For details about the test data set, the measurement environment and MLP-network see [3] and [4].

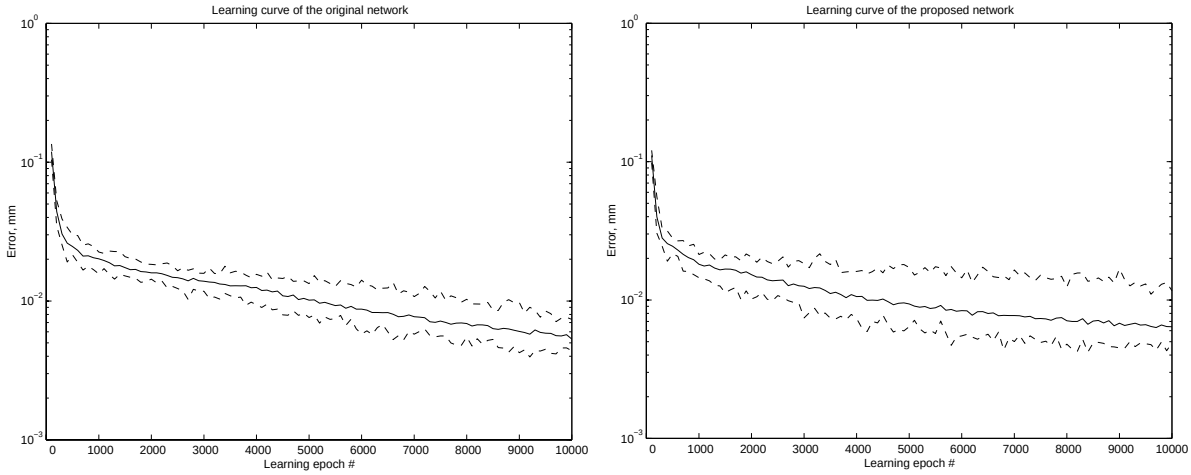

Figure 8. Learning curves of original and proposed network.

In Fig. 8 the learning curves for both configurations are presented. On both the mean value of the 20 test runs seems to behave quite similarly. However, looking at the minimum and maximum errors in training it can be seen that the variance in the original configuration (left side) is smaller than in the proposed configuration. This is most likely due to difference in the derivatives. The higher valued derivatives of the proposed network seem to cause slightly stronger variations in the network's weight parameters.

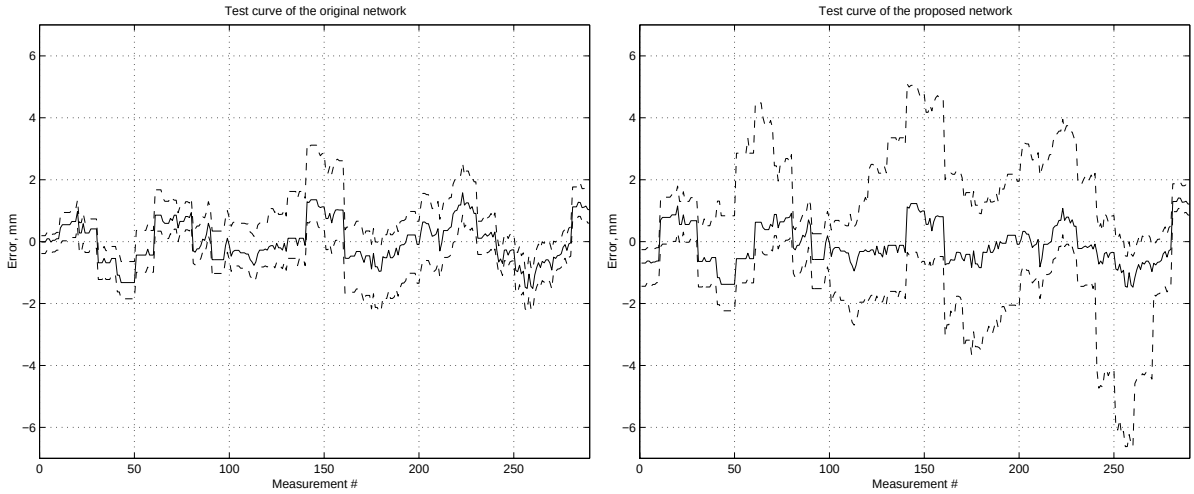

Figure 9. Test curves of original and proposed network.

Figure 9 shows the response of the network to the test data set. The curves represent the difference between the actual value and the output of the network. Again it can be seen that while the

mean values (solid lines) of the tests show similar kind of behaviour, the variance of error is larger in the proposed configuration.

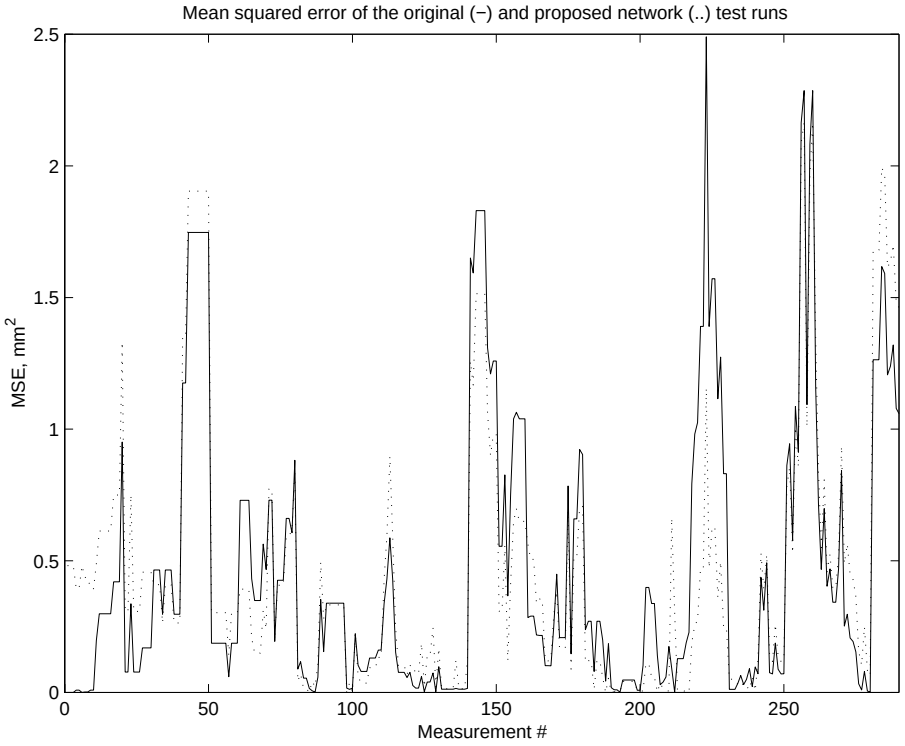

Figure 10. Squared error of the original (-) and proposed network (..).

Finally, in Figure 10, the mean squared error of 40 test runs is shown. The graphs show similar kind of behaviour for both networks. The mean of mean of squared errors of the 20 tests for the proposed network is 0.4601. Compared to the original network's corresponding value of 0.4669 the proposed network seems to have even slightly better performance. The sums of mean squared errors are 133.4418 and 135.4136 for the proposed and original configuration, respectively.

4 CONCLUSION

It is possible to use such activation functions in MLP-type of networks that do not utilise the traditional (exponential) type of activation functions. However, it is required for these activation functions to be continuous and continuously differentiable. In this paper it was shown that such functions can be made without any significant loss in the network performance. Even that there might be slightly more variation in training, it is possible to attain a network with acceptable performance by using the proposed functions.

REFERENCES

[1] Welstead, S.T., *Neural Network and Fuzzy Logic Applications in C/C++,* John Wiley & Sons, Inc., New York, 1994, 494 p.
[2] Haykin, S., *Neural Networks*, 2nd ed., Prentice Hall International, Inc., London, 1999, 842 p.
[3] Vehviläinen, P. V., Ihalainen, H. A. T., Jokinen, H. A., *Determining the Position of a Fabric by Neural Network*, IMEKO - XV, World Congress 1999, Osaka, Japan. Volume V, pp. 203-208.
[4] Vehviläinen, P., *Determining the position of a dryer fabric by soft computing*, Master of Science Thesis, Tampere, Tampere University of Technology, 1997, 56 p.

AUTHOR(S): Measurement and Information Technology, Tampere University of Technology, Box 692, FIN-33101, Tampere, Finland, Phone Int +358 3 365 3572, Fax Int +358 3 365 2171, E-mail: pekko@mit.tut.fi