

Approximation a One-Dimensional Functions by Using Multilayer Perceptron and Radial Basis Function Networks

Huda Dheyauldeen Najeeb
Department of public relations
College of Media, University of Al Iraqia, Baghdad, Iraq

Abstract- We present two methods of approximation a one-dimensional functions, first, by using Multilayer Perceptron (MLP) and second, by using Radial Basis Function networks (RBF). The two methods of Multilayer Perceptron and Radial Basis Function networks of feedforward network type are the same but they are different in relative to input and output. However, the Radial Basis Functions neural networks (RBF) compared to the MLP have an advantage that their training is much less computational powerful. In this paper, we calculated the number of training cycles which is needed to approximate a one-dimensional function by using these two methods. It was found that RBF has a faster training process than MLP it reaches to 65 times as well as the accuracy of RBF, as it is more accurate in spite of using more than one hidden layer in MLP therefore we conclude that the method of MLP is not in appropriate for making the approximate function thus we have to use the RBF method instead of it to approximate a one-dimensional functions.

Keywords: Multilayer Perceptron , Radial Basis Function networks , Exact interpolation, Gaussian Functions.

I. INTRODUCTION

The approximation of functions which are dealing with artificial neural networks. The theme of the approximation problem needed to be explained are the inputs and outputs, the minute details of the approximation had been selected and graded the top results which the inputs and outputs depend up on it. Multilayer Perceptron (MLP) and Radial Basis Functions neural networks (RBF) used more in common in artificial neural networks (Werbos [1] and Rumelhart [2]). Structurally, Radial Basis Functions neural network (RBF) has only one hidden layer reverse Multilayer Perceptron (MLP) has one or more hidden layers and the RBF network has a simpler structure and a faster training process than MLP [3], but how much faster, in other words, how much number of training cycles at the very least to an approximate a one-dimensional function using RBF compared to MLP to approximate the same function. It is precision hand: Are we going to get a better result if we use more than one hidden layer in MLP. Therefore we have selected one-dimensional function and approximated it using Multilayer Perceptron and Radial-Basis Functions, then analyze and compare the results.

II. PROPOSED ALGORITHM

A. Multilayer Perceptron networks

MLP is an abbreviation of the term "Multilayer Perceptron". It can be approximated well when putting big enough hidden layer with suitable coefficient values.[4] Their main features are (see figure.1): a) It's a Feedforward network type. b) Able to learning surfaces decision of not linear. c) Based on sigmoid unit with differentiable threshold function.

$$\theta(\lambda) = \frac{1}{1+e^{-\lambda}} \quad (1)$$

$$\lim_{\lambda \rightarrow +\infty} \theta(\lambda) = 1 \quad (2)$$

$$\lim_{\lambda \rightarrow -\infty} \theta(\lambda) = 0 \quad (3)$$

MLP structure and design

Despite of the emergence of different types of neural network models since 1940, but the Multilayer Perceptron (MLP) is still in use updatable (Mata, 2011). This network contains three layers called: input layer, output layer, and hidden layer is between them, every layer contain one or more neurons. [5]

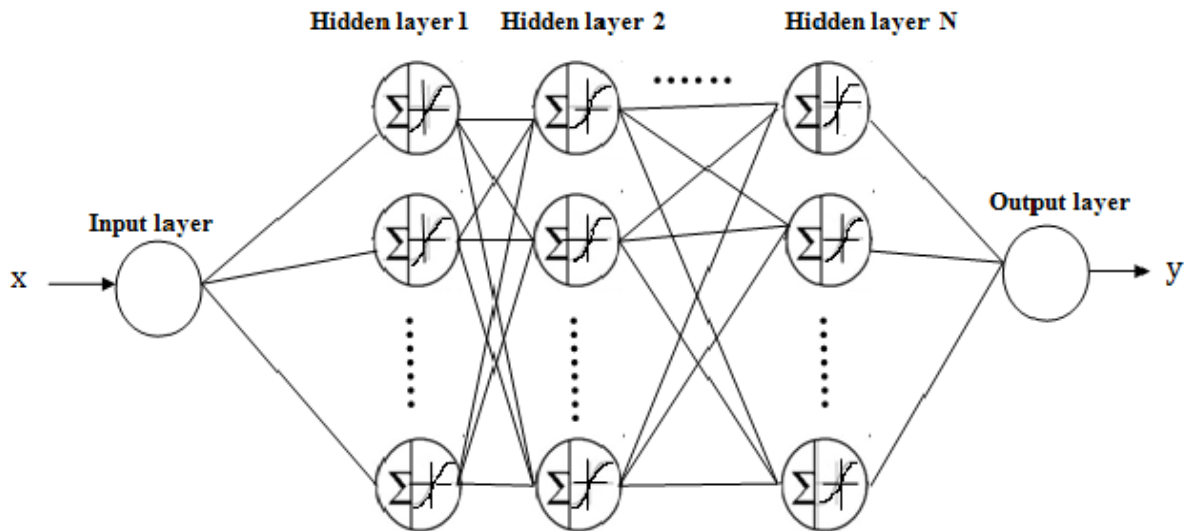


Figure.1.Representation of a MLP

If this network has two neurons in the hidden layer (as shown in figure.2) the output of the first neuron is

$$\frac{1}{1+e^{-(w_1x-\theta_1)}} \quad (4)$$

and the output of the second neuron is $\frac{1}{1+e^{-(w_2x-\theta_2)}} \quad (5)$

Then the total output of the neural network as:

$$\frac{1}{1+e^{-(w_3 \cdot \frac{1}{1+e^{-(w_1x-\theta_1)}} + w_4 \cdot \frac{1}{1+e^{-(w_2x-\theta_2)}} - \theta_3)}} \quad (6)$$

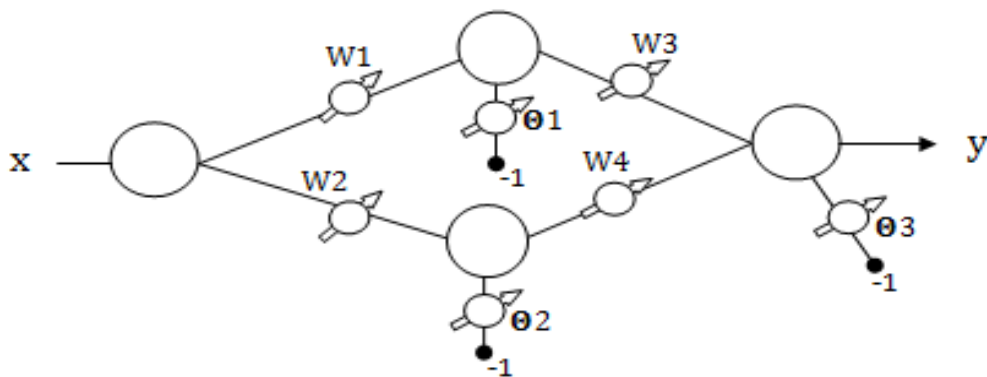


Figure .2.Representation of two neurons in MLP

The goal is to map an input vector x into an output $y(x)$. We can describe the layers as:

- a) The input layer: the data vector or pattern is accepted by the input layer.
- b) The hidden layers: from the previous layer the output are accepted, then weight them, ordinarily, not linear activation function.
- c) The output layer: The output is taken from the final hidden layer and possibly passes through an output not linearity, to get the required values.[6]

B. Radial Basis Function networks

Firstly, we have already explained the structure of Multilayer Perceptron (MLP) networks and seen how MLP can learn to approximate functions. Secondly, we will have a general look about another and different method named "Radial-Basis Functions neural networks (RBF)".

The best way of approximation is using RBF networks extensively in various fields of science and engineering applications (Benghanem and Mellit, 2010; Mellit and Kalogirou, 2008) as well as a high potential for approximation, simplicity of structure, learning algorithms faster and resulted from the theory of function approximation.[7]The basic specifications of (RBF) Networks are:

a)Its a feedforward network type.b) The application of sum of radial basis functions emerged clearly in the hidden nodes. c)The output nodes in this method like the output nodes in an MLP which implement linear summation functions. d)There are two steps of training: step one from the input layer the weights are specified then mapped to hidden layer, step two from the hidden layer the weights are specified then mapped to output layer. e) It is very fast method in the training/learning and very good at interpolation.

Radial Basis Functions

Experimental and theoretical studies refer to characteristics of the interpolating caused a relative insensitivity to the suitable shape of the basis functions $\Theta(x)$. The basis functions are as followed:[8]

❖ Multi quadric Functions:

$$\Theta(x) = (x^2 + \lambda^2)^{1/2} \text{ with parameter } \lambda > 0$$

❖ Gaussian Functions:

$$\Theta(x) = \exp\left(-\frac{x^2}{2\lambda^2}\right) \text{ with parameter } \lambda > 0$$

❖ Inverse Multi quadric Functions:

$$\Theta(x) = (x^2 + \lambda^2)^{-1/2} \text{ with parameter } \lambda > 0$$

❖ Generalized Multi quadric Functions:

$$\Theta(x) = (x^2 + \lambda^2)^\alpha \text{ with parameters } \lambda > 0, 1 > \alpha > 0$$

❖ Generalized Inverse Multi quadric Functions:

$$\Theta(x) = (x^2 + \lambda^2)^{-\beta} \text{ with parameters } \lambda > 0, \beta > 0$$

❖ Thin Plate Spline Function

$$\theta(x) = x^2 \ln(x)$$

❖ Linear Function:

$$\theta(x) = x$$

❖ Cubic Function:

$$\theta(x) = x^3$$

RBF Structure and design

The RBF neural network is a feedforward network type.[5] It also divided into three layers like the Multilayer Perceptron (MLP).These layers known: input layer, output layer, and hidden layer between them is a single. Each of these performs a radial basis function.

The RBF network has its origin in performing exact interpolation of a set of data points in a multi dimensional space. The exact interpolation means a particular situation as the outcome of the output function equal the data points in all precisely.[9]

Officially, the exact interpolation of a set of N data points in a multi-dimensional space demands all the M dimensional input vectors $x^\beta = \{x_j^\beta : j = 1, \dots, M\}$ referring to the conforming outputs α^β that is required.

Aiming to get a function $f(x)$ as:

$$f(x^\beta) = \alpha^\beta \quad \forall \beta = 1, \dots, N$$

Radial Basis Function (RBF) method presents a set of N basis functions, every "j" stand for data point takes the style $\theta(\|x - c_j\|)$ where is $\theta(\cdot)$ not linear function. Usually the distance $\|x - c_j\|$ taken to be Euclidean between x and c_j . The output of the mapping is a linear mixture of the basis functions.

$$f(x) = \sum_{j=1}^N w_j \theta(\|x - c_j\|) \quad (7)$$

The RBF is illustrated in figure.3:We identify a set of inputs and outputs (x_i, y_i) sequentially, and specify three various parameters called the location of the Gaussian kernels (C_i), the multiplicative factors (w_i), and their variances (Θ_i). [10,11] The approximation of y, by a RBF is known $f(x)$.The concerned approximation express the weight of N Gaussian kernels then the number of Gaussian kernels decided the complexity of this method.

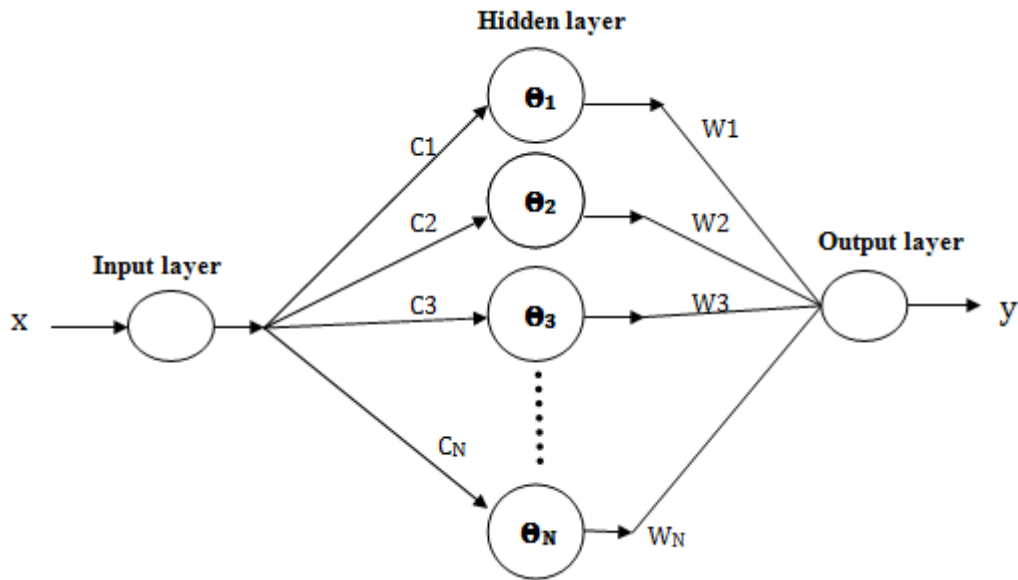


Figure .3..Representation of a RBF

III.EXPERIMENTAL RESULTS

In Matlab, we are selecting two different functions which are one-dimensional and approximate them by: Multilayer Perceptron(MLP) and Radial-Basis Functions neural networks (RBF).(see figure.4)

We selected three different configurations of both types of networks.Evaluate error for the training and test set depending on the numberof training cycles, and keep neural network and results for them.

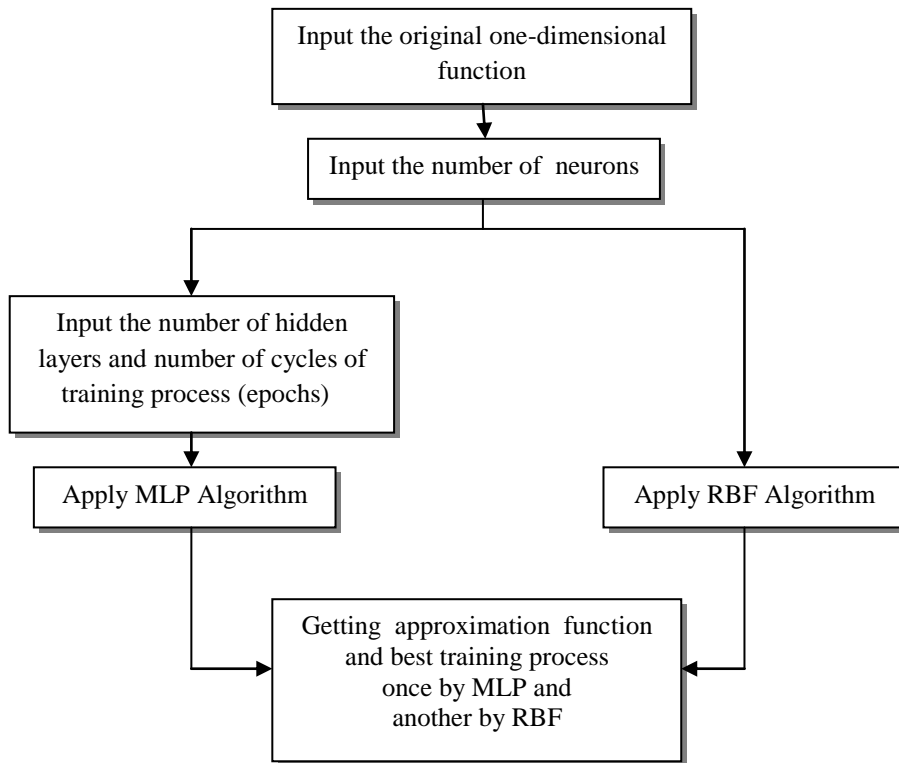
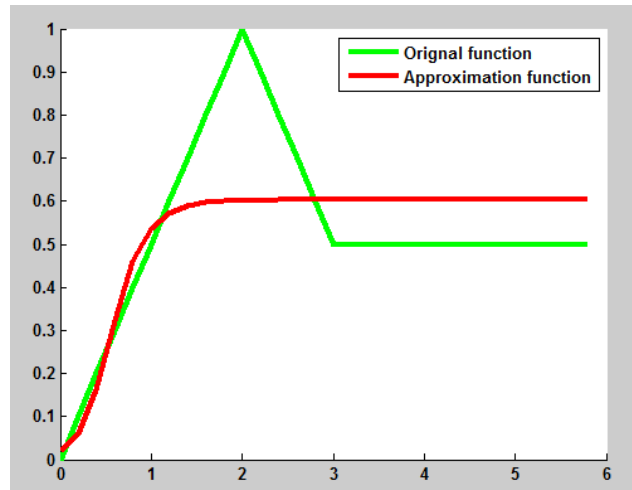
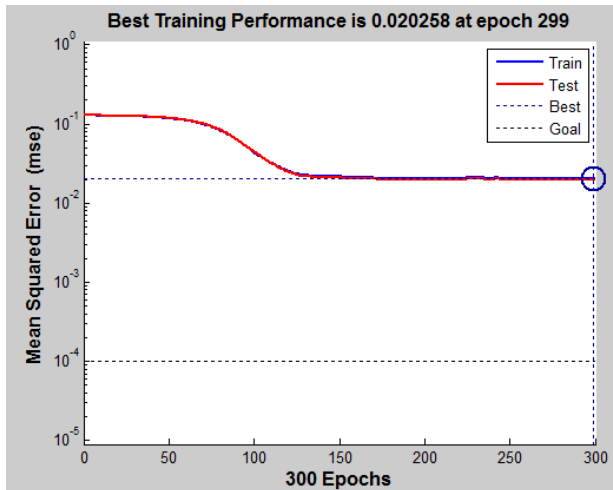


Figure .4.The propose work in Matlab

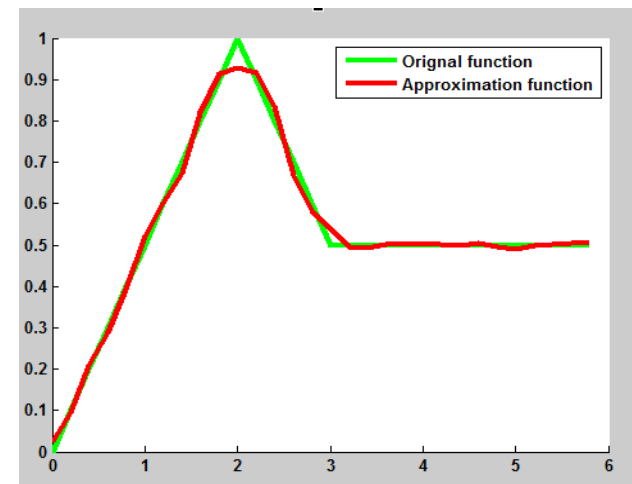
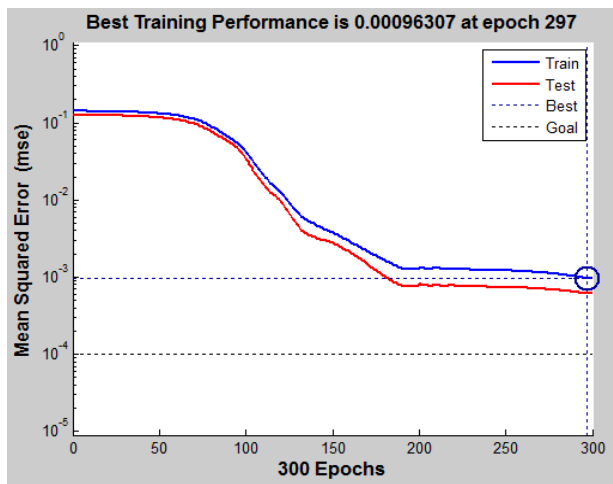
Approximation using Multilayer Perceptron

The function is been approximated by MLP method .Where:
 (Green color) represent the one-dimensional function which needed to be approximated, while (Red color) represent the result from the approximation process using this method (see figure.5 and figure.6).

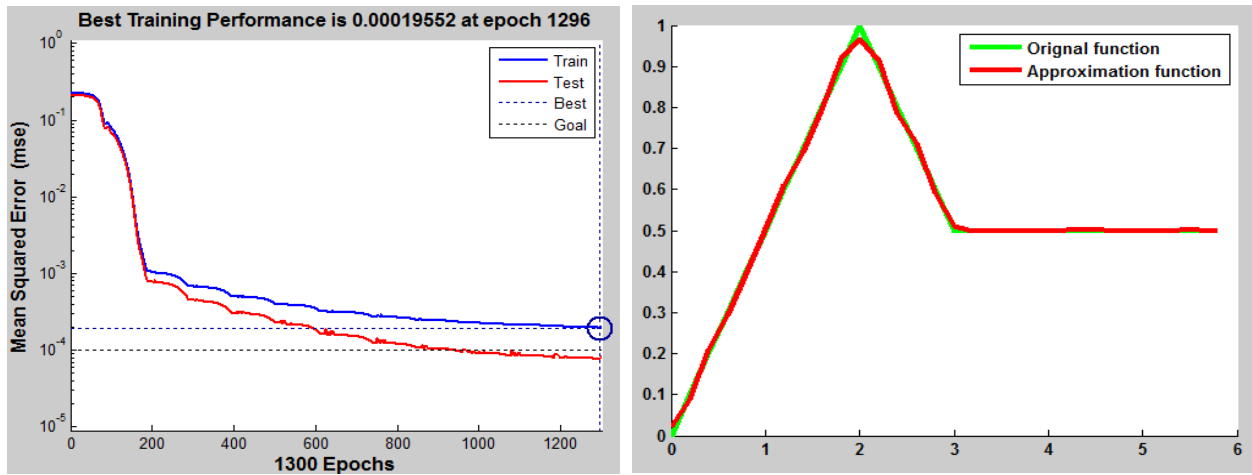
- a- At the beginning we create a simple network with two neurons in one hidden layer and 300 cycles of training process (epochs),we will observe the emergence of a clear error in approximation.
- b- Consequently, we have increased the number of neurons to ten neurons in the hidden layer with still the same number of cycles. The network approximated function more perfect.
- c- Gradually increasing the number of neurons, increasing the number of hidden layers and the number of cycles of training process we achieve better approximation properties. The training and test sample deviations gradually reduced.



a.Approximation using MLP with two neurons in hidden layer (300cycles)

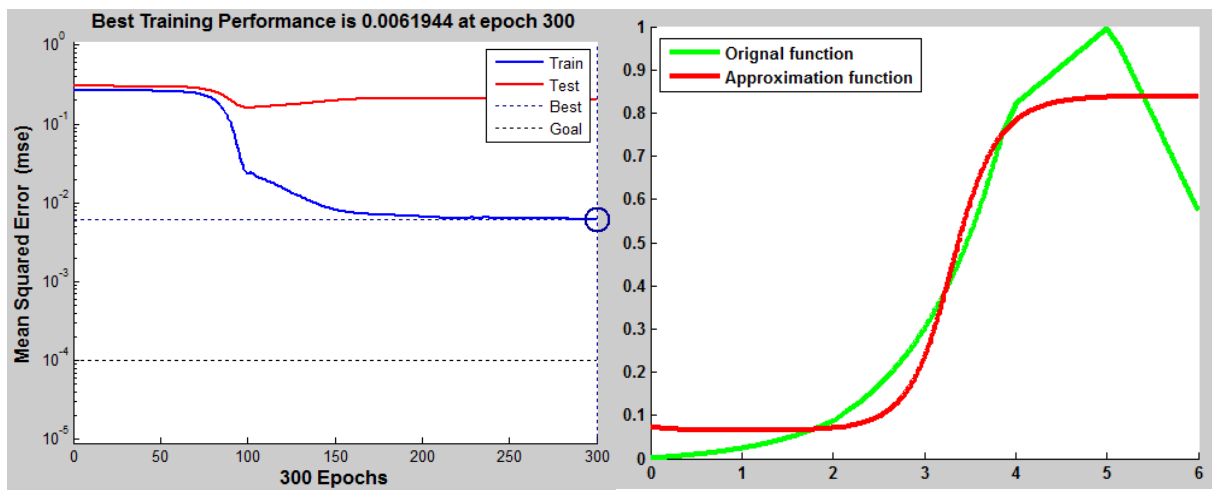


b.Approximation using MLP with ten neurons in the one hidden layer(300cycles)

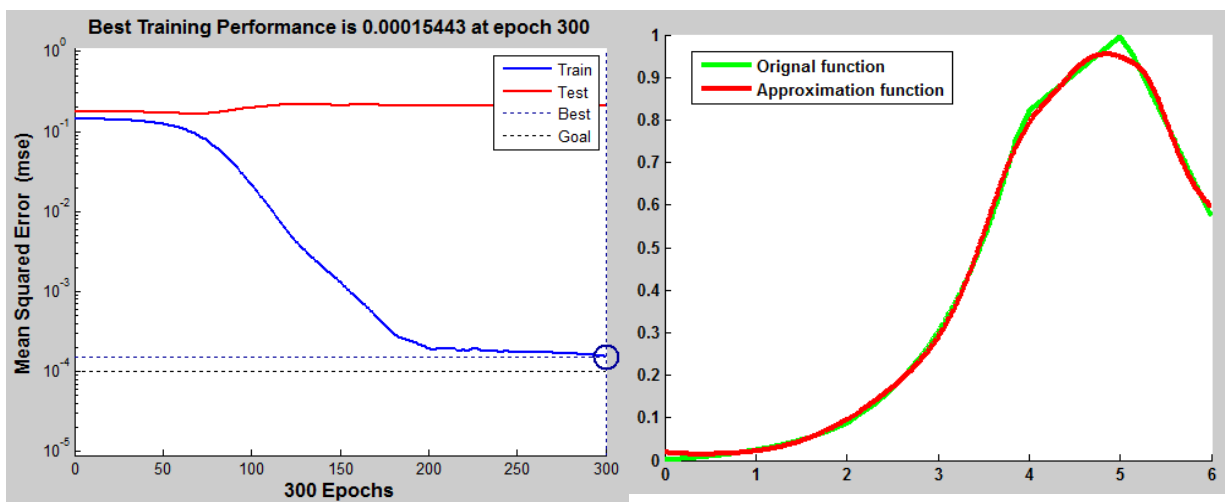


C. Approximation using MLP with twenty neurons in the four hidden layers (1300 cycles)

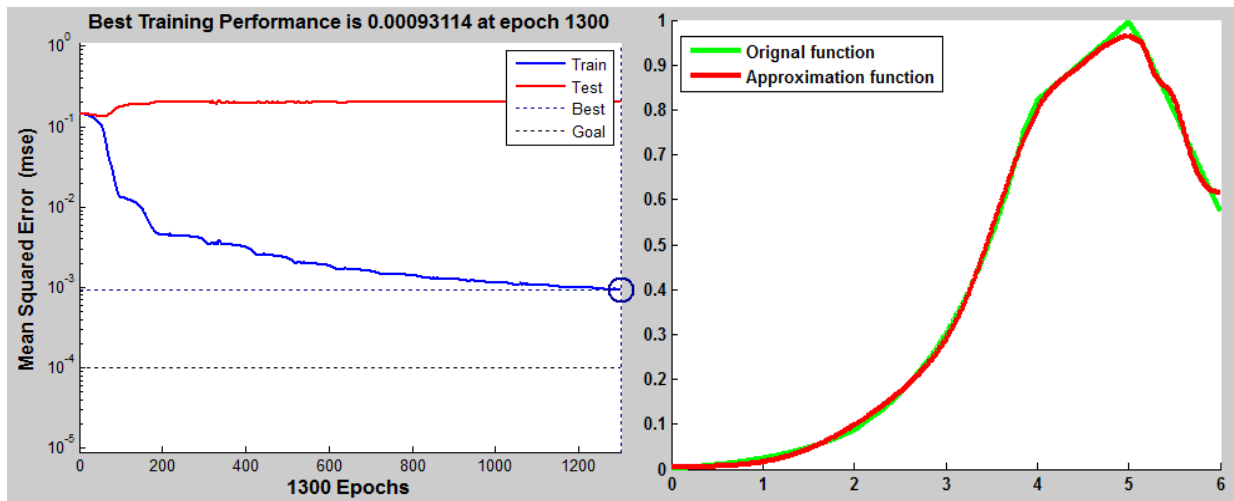
Figure .5. Approximation a one-dimensional function1 using MLP



a. Approximation using MLP with two neurons in hidden layer (300cycles)



b. Approximation using MLP with ten neurons in the one hidden layer(300cycles)



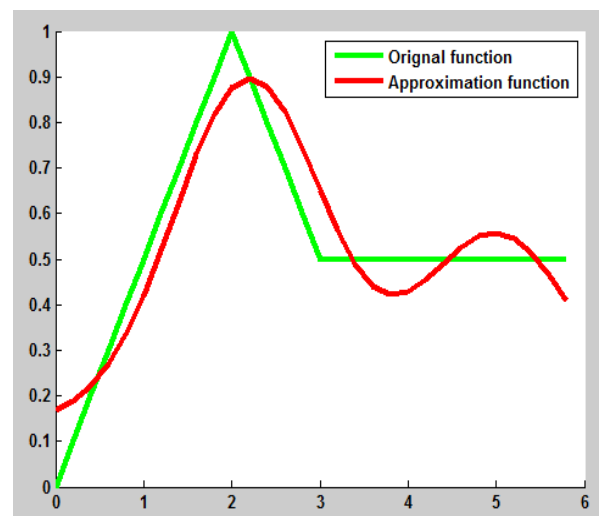
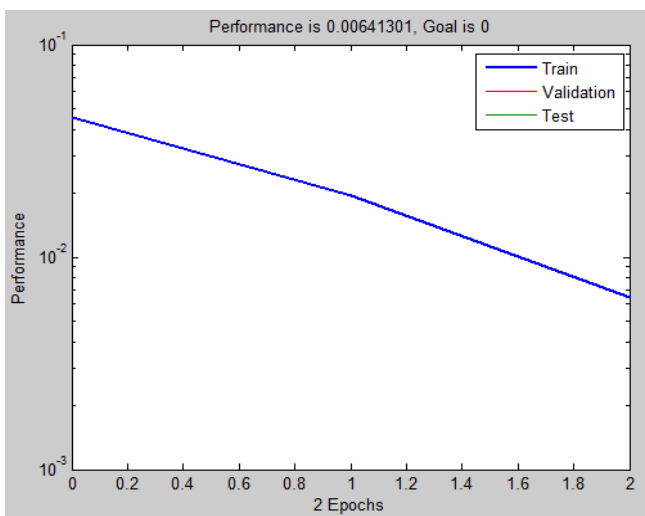
C. Approximation using MLP with twenty neurons in the four hidden layers (1300 cycles)

Figure .6. Approximation a one-dimensional function2 using MLP

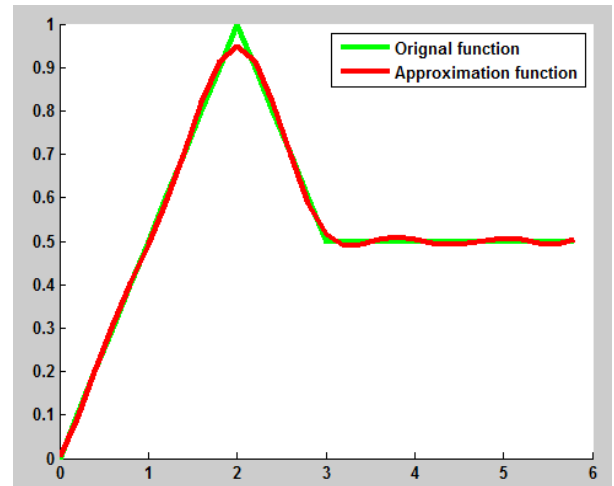
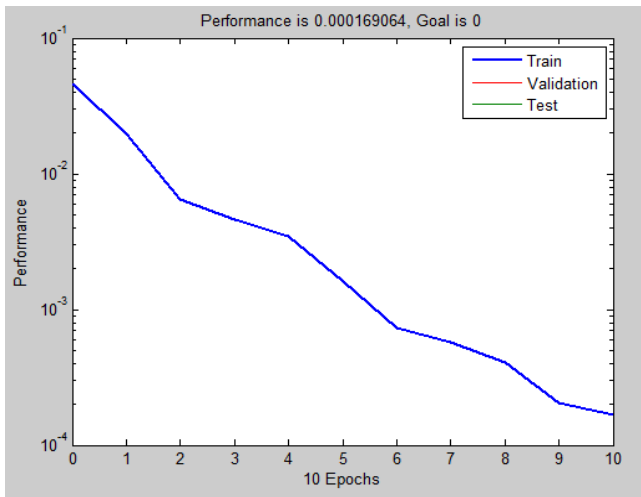
Approximation using Radial Basis Function neural networks (RBF)

The function is been approximated by RBF network method .Where we are taking the same number of neurons which was taken in the previous method , but with a single hidden layer and we will note that the approximation in this method is faster with better results.

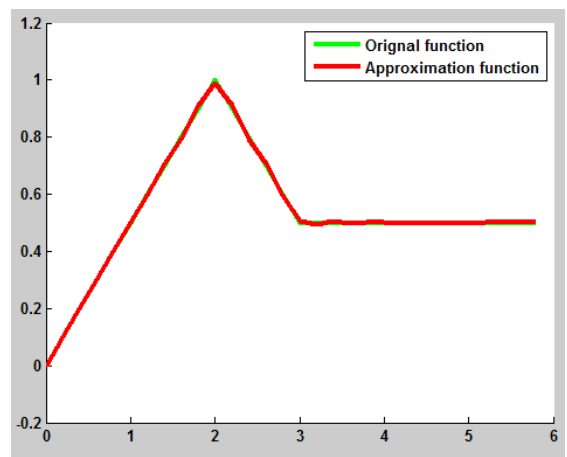
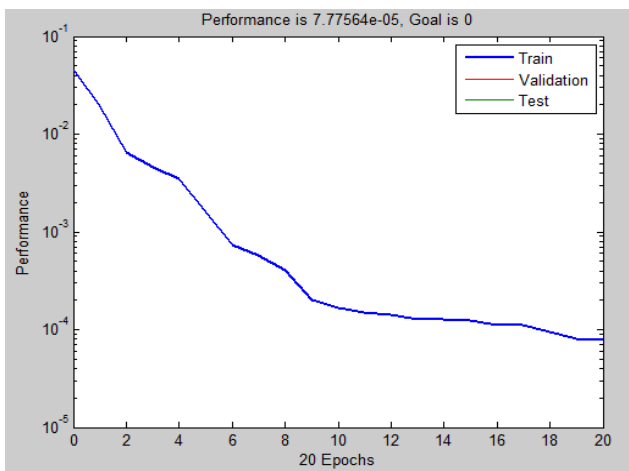
- a- At the beginning we choose the RBF network with two neurons and cycles of training process(epochs), neural network could approximate the original function but with significant errors.
- b- Subsequently, we used ten neurons with ten number of cycles of training process (epochs).This network has been more precisely approximates the original function.Compared with the previous method, RBF will faster the training process by 30-times and giving better result.
- c- At the last experiment, we conducted with twenty neurons and cycles of training process (epochs). The network in this case became more precise .Compared with the previous method, RBF will faster the training process by 65-times. In spite of the increase in hidden layers in MLP, but the result of BRF is remained the best(see figure.7 and figure.8).



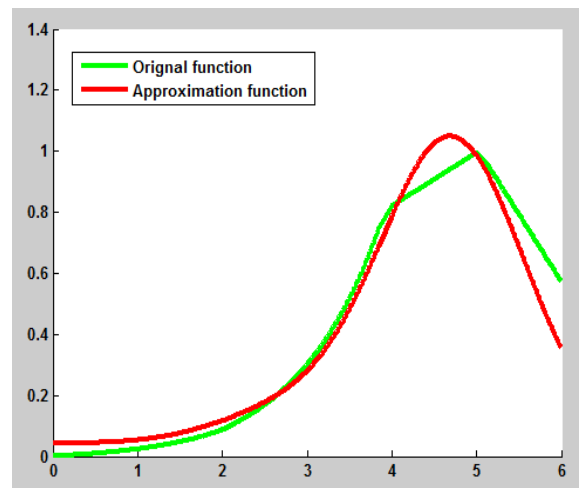
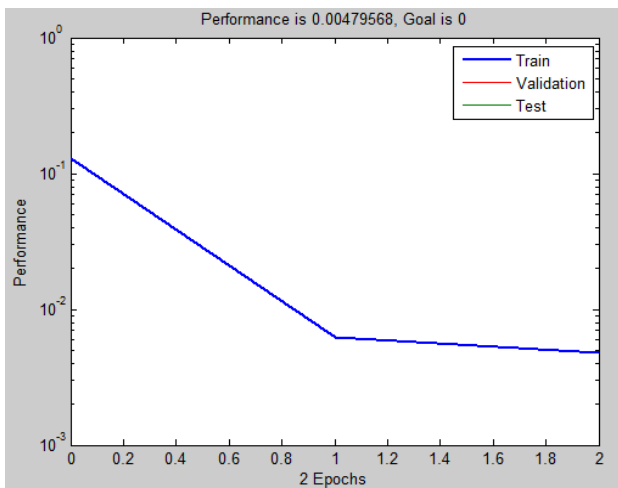
a. Approximation using RBF network with two neurons



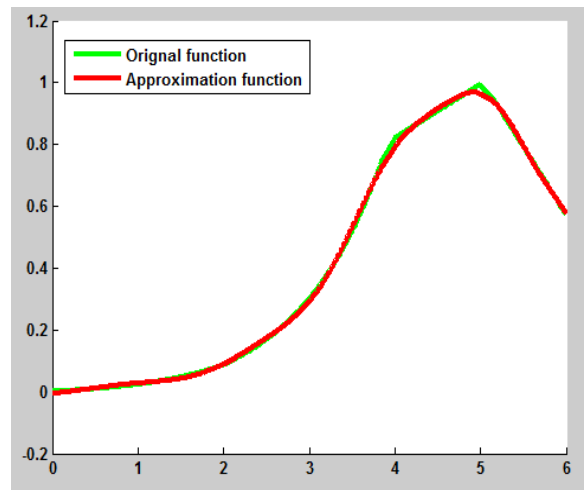
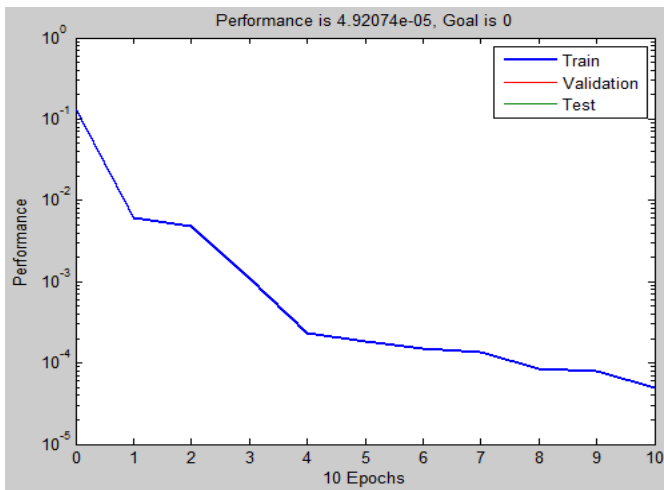
b. Approximation using RBF network with ten neurons



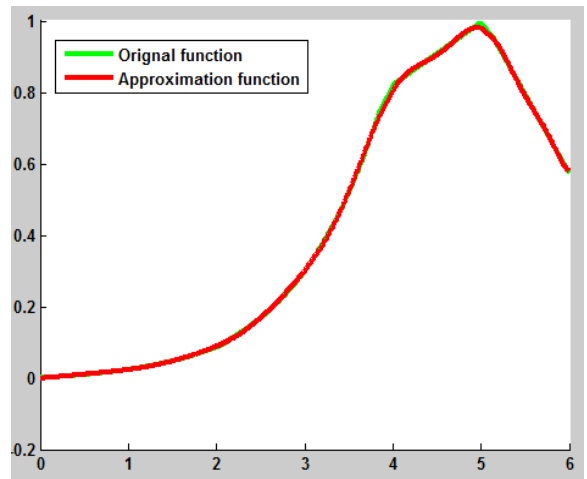
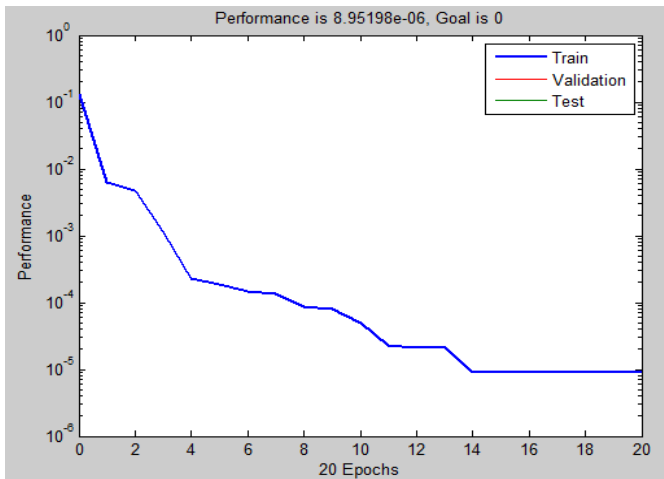
c. Approximation using RBF network with twenty neurons
Figure.7. Approximation a one-dimensional function using RBF network



a. Approximation using RBF network with two neurons



b. Approximation using RBF network with ten neurons



c. Approximation using RBF network with twenty neurons

Figure.8. Approximation a one-dimensional function₂ using RBF network

IV. CONCLUSION

Multilayer Perceptrons and Radial Basis Function networks are universal approximators and both of these methods are feedforward type network but with some differences include:

- 1) An MLP has one or more hidden layers, whereas an RBF network (In its simplest form) has one hidden layer.
- 2) In the MLP, the hidden and output layers used as a pattern classifier usually are not linear. Nevertheless, in the RBF network the hidden layer is not linear, while the output is linear. When we use the MLP to solve not linear regression problems, usually a linear layer for the output is the preferred selection.
- 3) The map of MLP not linear input/output structured global approximations. Form the other point of view "Gaussian function" declare the structure of not linear input/output local approximations which the not linearities used by RBF network.

The results of this work suggest that RBF method provides more accurate results although MLP used more than one hidden layer and it has a much faster training process by 65-times than the MLP method. RBF need a limit number of training cycles and limit neuron in one hidden layer to an approximate a one-dimensional

function. This property has made RBF method used as substitute for MLP method to approximate a one-dimensional function.

REFERENCES

- [1].Werbos P., "Beyond Regression: New Tools for Prediction and Analysis in The Behavioral Sciences", PhD thesis, Harvard University, 1974.
- [2].Rumelhart D.and Hinton G., Williams R, "Learning Representation by Back Propagating Errors", Nature 323, pp. 533-536, 1986.
- [3].Yue Wu,1 Hui Wang,1 Biaobiao Zhang,1 and K.-L. Du1," Using Radial Basis Function Networks for Function Approximation and Classification ", ISRN Applied Mathematics,2012.
- [4].Paavo Nieminen, " Classification and Multilayer Perceptron Neural Networks",Data Mining Course (TIES445), 2012.
- [5].Olanrewaju A Oludolapo and Adisa A Jimoh, Pule A Kholopane," Comparing Performance of MLP and RBF Neural Network Models for Predicting South Africa's Energy Consumption", Journal of Energy in Southern Africa , 23 , 3,pp. 42-45, August 2012.
- [6].Mark Gales, "Multi-Layer Perceptrons ", Module 4F10: Statistical Pattern Processing, University of Cambridge Engineering Part IIB,2015.
- [7].C. Lesage and M. Cottrell eds."Approximation By Radial Basis Function Networks", Connectionist Approaches in Economics and Management Sciences, Kluwer academic publishers, pp. 203-214, 2003.
- [8].Haykin, S., "Neural Networks: A Comprehensive Foundation", NY:IEEE Press, 1994.
- [9].John A. Bullinaria" Radial Basis Function Networks: Introduction Neural Computation: Lecture 13", 2015.
- [10].Verleysen M. and Hlavackova K., "An Optimised RBF Network for Approximation of Functions", ESANN, European Symposium on Artificial Neural Networks, Brussels (Belgium), pp. 175-180, 1994.
- [11].Benoudjit N., Archambeau C., Lendasse A.and Lee J., Verleysen M., "Width Optimization of The Gaussian Kernels in Radial Basis Function Networks", ESANN,European Symposium on Artificial Neural Networks, Bruges (Belgium), pp. 425-432,2002.