Function approximation using artificial neural networks

Article *in* WSEAS Transactions on Mathematics · June 2008

2 authors:

Zarita Zainuddin
Universiti Sains Malaysia

Pauline Ong
Universiti Tun Hussein Onn Malaysia

Function Approximation Using Artificial Neural Networks

ZARITA ZAINUDDIN & ONG PAULINE
School of Mathematical Sciences
Universiti Sains Malaysia
11800 Minden, Penang
MALAYSIA
zarita@cs.usm.my

Abstract: - Function approximation, which finds the underlying relationship from a given finite input-output data is the fundamental problem in a vast majority of real world applications, such as prediction, pattern recognition, data mining and classification. Various methods have been developed to address this problem, where one of them is by using artificial neural networks. In this paper, the radial basis function network and the wavelet neural network are applied in estimating periodic, exponential and piecewise continuous functions. Different types of basis functions are used as the activation function in the hidden nodes of the radial basis function network and the wavelet neural network. The performance is compared by using the normalized square root mean square error function as the indicator of the accuracy of these neural network models.

Key-Words: - function approximation, artificial neural network, radial basis function network, wavelet neural network

1 Introduction

Learning a mapping between an input and an output space from a set of input-output data is the core concern in diverse real world applications. Instead of an explicit formula to denote the function *f*, only pairs of input-output data in the form of *(x, f(x))* are available.

Let $$x_i \in R^m, \qquad i = 1,2,\ldots,N \qquad (1)$$
$$d_i \in R^1, \qquad i = 1,2,\ldots,N$$

be the N input vectors with dimension m and N real number output respectively. We seek an unknown function $f(x):R^m \rightarrow R^1$ that satisfies the interpolation where

$$f(x_i) = d_i, \qquad i = 1,2,\ldots,N \qquad (2)$$

The goodness of fit of d_i by the function f, is given by an error function. A commonly used error function is defined by

$$E(f) = \frac{1}{2}\sum_{i=1}^{N}(d_i - y_i)^2 \qquad (3)$$
$$= \frac{1}{2}\sum_{i=1}^{N}[(d_i - f(x_i))]^2$$

where y_i is the actual response.

In short, the main concern is to minimize the error function. In the other words, to enhance the accuracy of the estimation is the principal objective of function approximation.

There exist multiple methods that have been established as function approximation tools, where an artificial neural network (ANNs) is one of them. According to Cybenko [1] and Hornik [2], there exists a three layer neural network that is capable in estimating an arbitrary nonlinear function *f* with any desired accuracy. Hence, it is not surprising that ANNs have been employed in various applications, especially in issues related to function approximation, due to its capability in finding the pattern within input-output data without the need for predetermined models. Among all the models of ANNs, the multilayer perceptron (MLP) is the most commonly used.

Nevertheless, MLP itself has certain shortcomings. Firstly, MLP tends to get trapped in undesirable local minima in order to reach the global minimum of a very complex search space. Secondly, training of MLP is highly time consuming, due to the slow converging of MLP. Thirdly, MLP also fails to converge when high nonlinearities exist. Thus, these drawbacks deteriorate the accuracy of the MLP in function approximation [3, 4, 5].

To overcome the obstacles encountered by using an MLP, a radial basis function network (RBFN), which has been introduced by replacing the global activation function in MLP with a localized radial basis function, has been found to perform better than the MLP in function approximation [6,7].

The idea of combining wavelet basis functions and a three layer neural network has resulted in the wavelet neural network (WNN), which has a similar architecture with the RBFN. Since the first implementation of the WNN by Zhang and Benveniste [8, 9, 10], it has received tremendous

 Manuscript Received June 14, 2007; Revised December 24, 2007

attention from other researchers [11, 12, 13, 14] due its great improvement over the weaknesses of MLP.

This paper is organized as follows. In sections 2 and 3, a brief introduction of RBFN and WNN is presented. The types of basis functions used in this paper are given in section 4, while numerical simulations of both neural network models in function approximation are discussed in section 5, where the performance of RBFN and WNN are compared in terms of the normalized square root mean square error function. Lastly, conclusions are drawn in section 6.

2 Radial Basis Function Network

Radial basis function network was first introduced by Broomhead and Lowe in 1988 [6], which is just the association of radial functions into a single hidden layer neural network, such as shown in Figure 1.

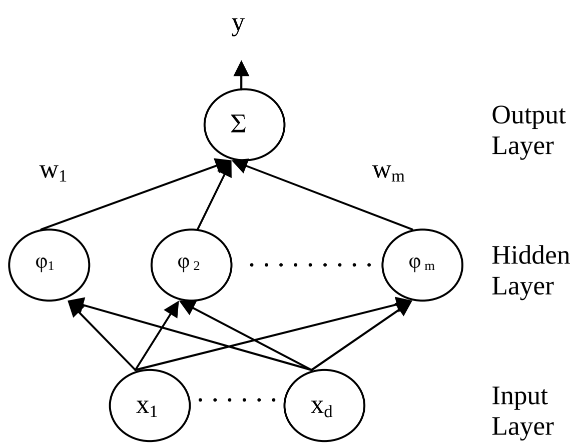


Figure 1: Radial basis function network

A RBFN is a standard three layer neural network, with the first input layer consisting of of d input nodes, one hidden layer consisting of m radial basis functions in the hidden nodes and a linear output layer. There is an activation function φ for each of the hidden node. Each of the hidden node receives multiple inputs $\vec{x} = (x_1,...,x_d)$ and produces one output y. It is determined by a center $\vec{c}$ and a parameter b which is called the width, where

$$\varphi_j(\xi) = \varphi_j(\|\vec{x} - \vec{c}\|/b), j = 1,...,m \ . \qquad (4)$$

φ can be any suitable radial basis function, such as Gaussian, Multiquadrics and Inverse Multiquadrics. Thus, the RBFN output is given by

$$y = \sum_{j=1}^{m} w_j \varphi_j(\xi) \qquad (5)$$

where $\varphi_j(x)$ is the response of the jth hidden node resulting from all input data, w_j is the connecting weight between the jth hidden node and output node, and m is the number of hidden nodes. The center vectors, $\vec{c}$, the output weights w_j and the width parameter b are adjusted adaptively during the training of RBFN in order to fit the data well.

2.1 Learning of Radial Basis Function Network

By means of learning, RBFN tends to find the network parameters c_i, b and w_i, such that the network output $y(x_i)$ fits the unknown underlying function $f(x_i)$ of a certain mapping between the input-output data as close as possible. This is done by minimizing an error function, such as in Equation 3.

The learning in RBFN is done in two stages. Firstly, the widths and the centers are fixed. Next, the weights are found by solving the linear equation. There are a few ways to select the parameter centers, c_i. It can be randomly chosen from input data, or from the cluster means. The parameter width, b, usually is fixed. Once the centers have been selected, the weights that minimize the output error are computed by solving a linear pseudoinverse solution.

Let us represent the network output for all input data, d, in Equation (5) as Y = Φ W where

$$\Phi = \begin{pmatrix} \varphi(x_1,c_1) & \varphi(x_1,c_2) & \ldots & \varphi(x_1,c_m) \\ \varphi(x_2,c_1) & \varphi(x_2,c_2) & \ldots & \varphi(x_2,c_m) \\ \vdots & \vdots & \vdots & \vdots \\ \varphi(x_d,c_1) & \varphi(x_d,c_2) & \ldots & \varphi(x_d,c_m) \end{pmatrix} \qquad (6)$$

is the output of basis functions φ, and

$$\varphi(x_i, c_j) = \varphi(\| x_i - c_j \|) \qquad (7)$$

Hence, the weight matrix W can be solved as

$$W = \Phi^{+}Y \qquad (8)$$

where Φ^{+} is the pseudoinverse defined as

$$\Phi^{+} = (\Phi^{T}\Phi)^{-1}\Phi^{T} \qquad (9)$$

3 Wavelet Neural Network

By incorporating the time-frequency localization properties of wavelet basis functions and the learning abilities of ANN, WNN has yet become another suitable tool in function approximation [8, 9, 10, 14].

WNN shares a similar network architecture as RBFN [15, 16], such as shown in Figure 1. Instead of the radial basis function, wavelet frames such as Gaussian wavelet, Morlet and Mexican Hat are used as the activation functions in the hidden nodes, while the center vectors and parameters of width in RBFN are replaced by translation $\overrightarrow{E_j}$ and dilation vectors $\overrightarrow{T_j}$ respectively. In fact, RBFN and WNN have been proven that they are actually specific cases of a generic paradigm, called Weighted Radial Basis Functions [15].

Similar to RBFN, WNN has a model based on Euclidean distance between the input vector $\vec{x}$ and translation vector $\overrightarrow{E_j}$, where each of the distance components is weighted by a dilation vector $\overrightarrow{T_j}$. Thus, the output of WNN is given by

$$y = \sum_{j=1}^{M} \Phi_j \left(\left\| \frac{\vec{X} - \vec{E}j}{\vec{T}j} \right\| \right) \tag{6}$$

where M is the number of hidden nodes.

Learning of WNN is similar to that of RBFN as discussed in section 2.1, where the parameters of the center and width of radial basis function in RBFN are replaced by the parameters of translation and dilation of the wavelet basis function in WNN respectively.

4 Types of Basis Function

In fact, RBFN and WNN are very similar to each other. The difference lies in the types of activation functions used in the hidden nodes of the hidden layer. Different types of basis functions that are used in this paper are given below (see Fig. 2):

i. Mexican Hat

$$\Phi(z) = \Phi\left(\left\| \frac{\vec{X} - \vec{E}}{\vec{T}} \right\| \right) = (n - 2z^2).\exp(-z^2); \tag{7}$$

where n = dim(z)

ii. Gaussian Wavelet

$$\Phi(z) = \Phi\left(\left\| \frac{\vec{X} - \vec{E}}{\vec{T}} \right\| \right) = (-z).\exp(-z^2/2) \tag{8}$$

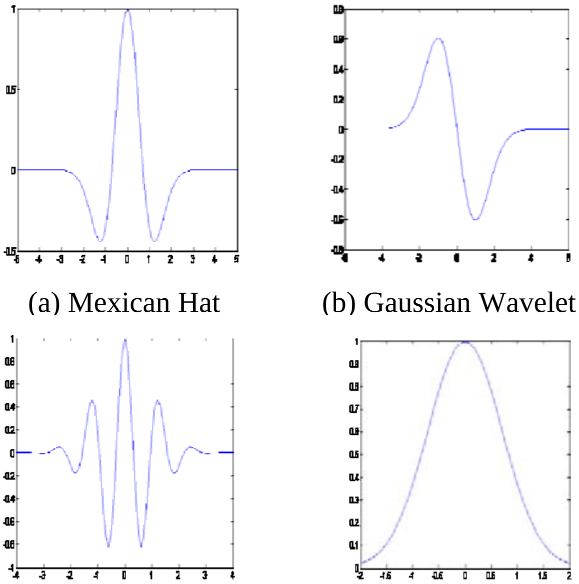
iii. Mexican Hat

$$\Phi(z) = \Phi\left(\left\| \frac{\vec{X} - \vec{E}}{\vec{T}} \right\| \right) = \cos(5z).\exp(-z^2/2) \tag{9}$$

iv. Gaussian

$$G(z) = G(\| x - c_i \|) = \exp(-z^2) \tag{10}$$

where $i = 1,2,..., N$



(a) Mexican Hat (b) Gaussian Wavelet

(c) Morlet (d) Gaussian

Figure 2: Types of basis functions

5 Numerical Simulations

In this section, we present experimental results of RBFN and WNN in approximating different types of functions. It includes a continuous function with one and two variables, and also a piecewise continuous function with one variable. Different types of basis functions will be used as the activation function in the hidden nodes of RBFN and WNN, namely, the Gaussian, Gaussian Wavelet, Morlet and Mexican Hat. The simulation is done by using Matlab Version 7.0 [17].

To evaluate the approximation results, an error criterion is needed. The normalized square root mean square error function (N_e) is chosen as the error criterion, that is,

$$N_e = \frac{1}{\sigma_y} \sqrt{\sum_{i=1}^{nt} (y^{(i)} - f^{(i)})^2} \tag{11}$$

where $f^{(i)}$ and $y^{(i)}$ are the output of network and desirable value of the function to be approximated , nt is the total number of testing samples and σ_y is the standard deviation of the output value. A smaller N_e indicates higher accuracy.

For function approximation of continuous functions with one variable, as in Case 1 and Case 2, these functions were sampled at 100 uniformly spaced training points in the domain [-1,1], while the testing samples consists of 200 points, which were also sampled uniformly in the same domain.

For the continuous functions with two variables, as in Case 3 and Case 4, 200 training points and 350 testing points in the domain [-1,1] were used.

Lastly, for the piecewise continuous function as in Case 5, we sample the function to yield 200 points distributed uniformly over [-10, 10] as training data. For testing data, 200 points which were uniformly sampled over [-10, 10] were used.

5.1 Case 1: 1-D Continuous Exponential Function

$$y = (x+1)\exp(-3x+3) \qquad (12)$$

The simulation result for Case 1 by using RBFN and WNN with different basis functions is shown in Table 1, in terms of N_e. Fig. 3 shows the results for the approximation of the function in Case 1.

Table 1: Simulation Result for Case 1

Model	Basis Function	N_e
RBFN	Gaussian	2.64481e-005
WNN	Gaussian Wavelet	0.0210695
	Mexican Hat	9.41899e-005
	Morlet	4.11055e-005

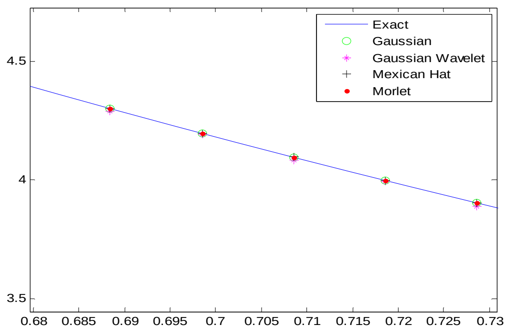


Figure 3: Simulation result for Case 1-zoom in

From Table 1, RBFN with Gaussian as the basis function gives the best performance. It is stated in [16] that RBFN approximates an exponential function well. Hence, in Case 1 where a 1-D exponential function is used, RBFN outperforms WNN. The wavelet basis functions used have the same characteristics, namely, they are crude, symmetric and irregular in shape, compactly supported, which means they vanish outside a finite interval and have explicit expression. However, among all the wavelet basis functions used, Gaussian wavelet yields the lowest accuracy. It is probably due to the shape of Gaussian wavelet, which is less similar to the function used in Case 1. Hence, it could not adapt well to the shape of this exponential function.

5.2 Case 2: 1-D Continuous Periodic Function

$$y = \sin(4\pi x)\exp(-|5x|) \qquad (13)$$

The simulation result for Case 2 by using RBFN and WNN with different basis functions is shown in Table 2, in terms of N_e. Fig. 4 shows the results for the approximation of the function in Case 2.

Table 2: Simulation Result for Case 2

Model	Basis Function	N_e
RBFN	Gaussian	9.11658
WNN	Gaussian Wavelet	0.207205
	Mexican Hat	6.09464
	Morlet	3.98573

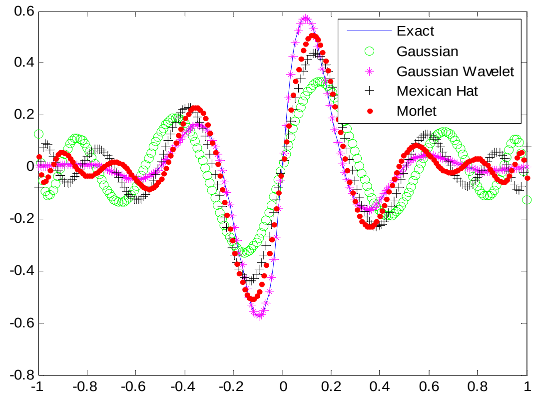


Figure 4: Simulation result for Case 2

From Table 2, WNN with Gaussian wavelet as the basis function gives the best performance. In Case 2 a 1-D periodic function is used. A periodic function is approximated better by WNN with an oscillating wavelet basis function [16]. Hence, it is shown in Table 2 that RBFN gives the lowest accuracy. Among all the wavelet basis functions, Gaussian wavelet performs the best. The good performance of Gaussian wavelet, as demonstrated in Fig. 4, is probably due to its similar shape with the function used in Case 2.

5.3 Case 3: 2-D Continuous Exponential Function

$$z = 2\sin(\pi\, e^{-x^2-y^2}) \qquad (14)$$

The simulation result for Case 3 by using RBFN and WNN with different basis functions is shown in Table 3, in terms of N_e.

Table 3: Simulation Result for Case 3

Model	Basis Function	N_e
RBFN	Gaussian	0.117956
WNN	Gaussian Wavelet	0.194523
	Mexican Hat	0.17059
	Morlet	0.15684

From Table 3, it can be observed that the accuracy of WNN with different types of basis function are comparatively close to each other. However, RBFN outperforms WNN since the function used in Case 3 is an exponential function.

5.4 Case 4: 2-D Continuous Periodic Function

$$z = 2(1-x^2-y^2)e^{-x2-y2} + \sin[(x^2+y^2)e^{-(x2-y2)/2}] \quad (14)$$

The simulation result for Case 4 by using RBFN and WNN with different basis functions is shown in Table 4, in terms of N_e.

Table 4: Simulation Result for Case 4

Model	Basis Function	N_e
RBFN	Gaussian	0.26292
WNN	Gaussian Wavelet	0.254194
	Mexican Hat	0.174942
	Morlet	0.114205

It is shown in Table 4 that WNN with Morlet as the basis function performs the best in comparison to the others. It could probably due to the shape of Morlet which is more identical to the shape of the function in Case 4. WNN outperforms RBFN in estimating a periodic function due to the oscillating behavior of the wavelet basis function which can capture the characteristic of a periodic function well. Hence, RBFN gives the highest error in Case 4.

5.5 Case 5: Piecewise Continuous Function

A piecewise continuous function

$$f(x)\begin{cases} -2.186x-12.864 & -10\le x<-2 \\ 4.264x & -2\le x<0 \\ 10e^{-0.05x-0.5}\sin[(0.03x+0.7)x] & 0\le x\le 10 \end{cases} \quad (15)$$

is used in this case. Simulation result in terms of N_e is shown in Table 5 while the results of approximation of the piecewise continuous function above are given in Figs. 5 and 6.

Table 5: Simulation Result for Case 5

Model	Basis Function	N_e
RBFN	Gaussian	0.899789
WNN	Gaussian Wavelet	2.71425e-013
	Mexican Hat	0.752215
	Morlet	0.650081

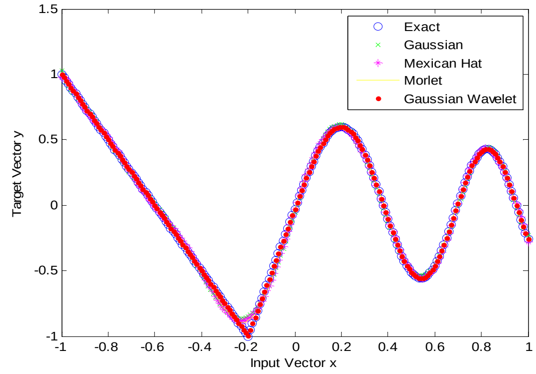


Figure 5: Simulation result for Case 5

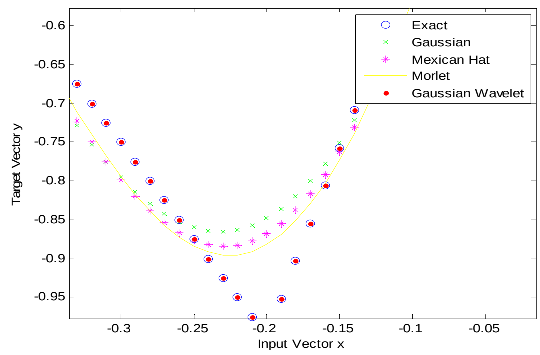


Figure 6: Simulation result for Case 5-zoom in

The function used in Case 5 is a piecewise continuous function, with a sharp spike at the interval (-0.3,-0.1). It is shown in Table 5 that WNN outperforms RBFN, especially in approximating the sharp spike, as shown in Figure 6. It is due to the irregular shape and fast oscillating characteristics of the wavelet functions, which leads them to approximate a function with sharp changes efficiently. Among all the wavelet basis functions, Gaussian wavelet gives the best performance. It is probably due to the shape of the Gaussian wavelet which is more identical to the piecewise function used in Case 5.

5.6 Discussion

An exponential function and a periodic function are approximated more accurately with RBFN and WNN respectively. It is mainly due to the basis functions that are used in the hidden nodes. WNN with an "oscillating" wavelet function tends to capture the behavior of a periodic function better due to its oscillating characteristic. Hence, in Case 2 and Case 4 which involve the approximation of a periodic function, it is observed that wavelet basis function outperforms radial basis function in terms of accuracy. However, in Case 1 and Case 3, where an exponential function is used, the accuracy of WNN and RBN is vice versa.

The function used in Case 5 is a piecewise continuous function. From the simulation result, it is observed that WNN shows a higher accuracy than RBFN, especially in approximating the sharp spike, which is shown in Figure 6. It is due to the irregular shape of the wavelet functions, which leads them to analyze function with discontinuity or sharp changes efficiently.

The wavelet basis functions used in this paper have the same characteristics, where they are crude, symmetric and have explicit expression.

6 Conclusion

This paper presents function approximation by using radial basis function network and wavelet neural network. WNN performs better in approximating a periodic function, whereas RBFN yields higher accuracy in estimating exponential function. In capturing the sharp spike in a piecewise function, WNN outshines RBFN due to its intrinsic characteristic.

References:

[1] Cybenko, G., Approximation by Superposition of a Sigmoidal Function, *Mathematics Control Signals Systems*, Vol. 2, 1989, pg 303-314.

[2] Hornik, K., M. Stinchcombe & H. White, Multilayer Feedforward Networks are Universal Approximator, *Neural Networks*, Vol. 2, 1989, pg 359-3661.

[3] M.H. Ahmad Fadzil & Zarita Zainuddin, Optimization of Learning Methods for Face Recognition using Multilayer Perceptron, Proceedings of the 3rd IASTED Int. Conf. On Signal and Image Processing, Honolulu, Hawaii, USA, Aug 13-16 2001.

[4] Zarita Zainuddin & Evans, D.J., "Human Face Recognition using Accelerated Multilayer Perceptrons", *International Journal of Computer Mathematics*, Vol. 80, No.5, pg 535-558

[5] Haykin,S., *Neural Network: A Comprehensive Foundation*, New York: Mc Millan Publishing Company, 1994

[6] Broomhead, D.S., Lowe, D,. Multivariable function interpolation and adaptive networks. *Complex Systems* , Vol. 2, 1988, pg 321-335.

[7] J.Moody, C. Darken, Learning with localized receptive fields, in: D.Touretzky, G.Hinton, T.Sejnowski (Eds), Proceedings of the 1988 Connectionist Models Summer School, San Mateo, CA, 1989

[8] Q. Zhang, Using Wavelet Network in Nonparametric Estimation, *IEEE Transactions on Neural Networks*, Vol.8, No.2, 1997, pg227-236

[9] Q.Zhang, A. Benveniste, Wavelet Networks, *IEEE Transactions on Neural Networks*, Vol 3, No. 6, 1992, pg889-898

[10] Q.Zhang, A. Benveniste, "Approximation by Nonlinear Wavelet Networks", *IEEE Transactions on Neural Networks*, Vol 6, No. 10, 1991, pg 3417-3420.

[11] A. Subasi, A. Alkan, E. Koklukaya & M.K. Kiymik, "Wavelet Neural Network Classification of EEG Signals by Using AR Model with MLE Preprocessing", Elsevier on Neural Networks Vol.18, 2005, pg 985-997.

[12] Abdulhamit Subasi, Epileptic Seizure Detection Using Dynamic Wavelet Networ, *Elsevier on Expert Systems with Application*, Vol. 29, 2005, pg 343-355.

[13] Eric A. Rying, Focused Local Learning With Wavelet Neural Networks, *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, 2002, pg 304-319.

[14] Wang Ting & Yasuo Sugai, Wavelet Neural Network for the Approximation of Nonlinear Multivariable Function, *IEEE Transactions on Neural Network*, 1999, pg 378-383.

[15] L.M. Reyneri, Unification of Neural and Fuzzy Computing Paradigms, in Proceedings of AT-96, 1st International Symposium on Neuro-Fuzzy Systems, Lausanne, 1996.

[16] V.Colla, M.Sgarbi, L.M. Reyneri, A Comparison between weighted radial basis functions networks and wavelet networks, Proceedings ESANN, 1998), p13-19

[17] Matlab 7 Release 14, *Neural Network Toolbox*, The MathWork.