On Comparison Between Radial Basis Function and Wavelet Basis Functions Neural Networks

L.N. M. Tawfiq, T.A. M.Rashid
Department of Mathematics ,College of Education Ibn Al-Haitham ,
Universityof Baghdad

Abstract

In this paper we study and design two feed forward neural networks. The first approach uses radial basis function network and second approach uses wavelet basis function network to approximate the mapping from the input to the output space. The trained networks are then used in an conjugate gradient algorithm to estimate the output. These neural networks are then applied to solve differential equation. Results of applying these algorithms to several examples are presented.

1. Introduction

Neural networks are connectionist models proposed in an attempt to mimic the function of the human brain. A neural network (Ann) consists of a large number of simple processing elements called neurons (or nodes) [1], [2]. Neurons implement simple functions and are massively interconnected by means of weighted interconnections. These weights, determined by means of a training process, determine the functionality of the neural network. The training process uses a training database to determine the network parameters (weights).

The functionality of the neural network is also determined by its topology. Most networks have a large number of neurons, with the neurons arranged in layers. In addition to input and output layers, there are usually layers of neurons that are not directly connected to either the input or the output, called hidden layers. The corresponding nodes are referred to as hidden nodes. Hidden layers give the network the ability to approximate complex, nonlinear functions.

The advantages of using neural networks are numerous: neural networks are learning machines that can learn any arbitrary functional mapping between input and output, they are fast machines and can be implemented in parallel, either in software or in hardware [3]. In fact , the computational complexity of Ann's is polynomial in the number of neurons used in the network. Parallelism also brings with it the advantages of robustness and fault tolerance. Efficient learning algorithms ensure that the network can learn mappings to any arbitrary precision in a short amount of time. Furthermore, the input-output mapping is explicitly known in a neural network and gradient descent procedures can be used advantageously to perform the inversion process.

2. Radial Basis Function Neural Networks

Radial basis function neural networks (RBFNN) are a class of networks that are widely used for solving multivariate function approximation problems [5],[4]. An RBFNN consists of an input and output layer of nodes and a single hidden layer. Each node in the hidden layer implements a basis function and the number of hidden nodes is equal to the number of points in the training database. The RBFNN approximates the unknown function that maps the input to the output in terms of a basis function expansion, with the functions, $G(x, x_i)$ as the basis functions. The input-output relation for the RBFNN is given by : $F(x) =$

$$\sum_{j=1}^{H} W_{\ell j} \exp\left(-\frac{\| x - c_j \|^2}{2\sigma_j^2}\right)$$

Where H is the number of basis functions used, $y = (y_1, y_2, \ldots, y_M)^T$ is the output of the RBFNN, **x** is the test input, x_j is the center of the basis function and are the expansion coefficients or weights associated with each basis function. Each training data sample is selected as the center of a basis function. Basis functions $G(x, x_i)$ that are radially symmetric are called radial basis functions. Commonly used radial basis functions include the Gaussian and inverse multiquadrics.

3. Wavelet Basis Function Neural Networks (WBFNN)

The wavelet transform is a time-frequency transform that provides both the frequency as well as time localization in the form of a multi resolution decomposition of the signal [6]. Consider a square - integrable function F(x) and let V_m be the vector space containing all possible projections of *F* at the resolution m where 2^m is the sampling interval at this resolution [7]. Obviously, as m increases, the number of samples at that resolution decreases and the approximation gets coarser. Now, consider all approximations of *F* at all resolutions. The associated vector spaces are nested as follows : $\ldots\subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset\ldots$ due to the fact the finer resolutions contain all that required information to compute the coarser approximation of the function *F*. It is also obvious that as the resolution decreases, the approximation gets coarser and contains less and less information. In the limit, it converges to zero:

$$\lim_{m\to\infty} V_m = \bigcap_{m=-\infty}^{\infty} V_m = \{0\}$$

On the other hand, as the resolution increases, the approximation has more information and eventually converges to the original signal :

$$\lim_{m\to-\infty} V_m = \bigcup_{m=-\infty}^{\infty} V_m \quad \text{is dense in } L^2(R).$$

If $\phi(x)$ denotes the scaling function, then V_m = linear span $\{\phi_{mk}, k\in Z\}$

where $\phi_{mk} = \sqrt{2^{-m}}\ \phi(2^{-m}x - k)$, $(m, k) \in Z^2$ is the translated version of $\phi(x)$.

Since the family of functions $\{\phi_{mk}(x) \mid (m, k) \in Z^2\}$ forms an orthonormal basis for V_m , F can be written as :

$$F_m(x) = \sum_{k=-\infty}^{\infty} s_{mk}\,\phi_{mk}(x) \quad \text{where} \quad s_{mk} = \int_{-\infty}^{\infty} F(x)\,\phi_{mk}(x)\,dx$$

is the projection of *F* onto the orthonormal basis functions $\phi_{mk}(x)$.

Further, suppose W_m is the orthogonal complement of V_m in V_{m-1}. Then

$V_{m-1} = V_m \oplus W_m$ with $V_m \perp W_m$ …….(1)

The $(m-1)^{th}$ approximation can be written as the sum of the projections of F onto V_m and W_m . Equivalently, the difference in information (called the dilates) between the m^{th} and $(m-1)^{th}$ approximations is given by the projection of F onto W_m. Mallat [8] shows that there exists a unique function, called the wavelet function, whose translates and dilates form an orthonormal basis for the space W_m. In other words, the detail of F at the m^{th} resolution is given by

$$D_m F(x) = \sum_{k=-\infty}^{\infty} d_{mk}\,\psi_{mk}(x) \quad \text{, where } \psi(x) \text{ is the wavelet ‘}$$

$\Psi_{mk}(x) = \sqrt{2^{-m}}\ \psi(2^{-m}x - k)$, $(m, k) \in Z^2$ are the translates and dilates of $\psi(x)$ and

$d_{mk} = \int_{-\infty}^{\infty} F(x)\psi_{mk}(x)dx$ are the projections of F onto W_m.

Further from (1), we get $F_{m-1}(x) = F_m(x) + \sum_{k=-\infty}^{\infty} d_{mk}\psi_{mk}(x)$.

Since the V-spaces form a nested set of subspaces, F can be written as :

$$F(x) = \sum_{k=-\infty}^{\infty} s_{k,-\infty}\, \phi_{k,-\infty}(x) + \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{lk}\, \psi_{lk}(x) \text{ ....(2)}$$

where l indexes over the different resolutions. In practice, the limits of summation are chosen to be finite. The architecture network consists of an input and an output layer with a single hidden layer of nodes [9]. The hidden layer nodes are grouped by resolution level. We have as many groups as resolution levels, with the number of basis functions at each resolution . The input-output relation is given by :

$$y_l = \sum_{j=1}^{H_1} w_{lj}\, \phi_j (x ,c_j) + \sum_{n=1}^{L} \sum_{k=1}^{K_n} w_{lkn}\, \psi_{nk}(x ,c_{nk}) \text{ ....(3)}$$

where L is the total number of resolutions H_1 is the number of scaling functions used at the coarsest resolution, K_n is the number of wavelet functions used at resolution *n*, $\mathbf{c}_i$ is the center of the corresponding basis function and w_{lj} is the weight of the interconnection connecting the j^{th} hidden node to the l^{th} output node. The weights are determined in a similar manner to the weights in the RBFNN described earlier.

The primary advantage of using wavelet basis functions is orthonormality. Orthonormality of wavelets ensures that the number of basis functions required to approximate the function F is minimum. The second advantage is that wavelets are local basis functions (localization property of wavelets). The multi resolution approximation (MRA) using wavelets allows the distribution of basis functions based on the resolution required in different parts of the input space. In addition, the ability to add details at higher resolutions as more data become available allows the network to learn in incremental fashion and allows the user to control the degree of accuracy of the approximation. Equation (2) formulated for scalar inputs can be extended for multidimensional inputs. The corresponding multidimensional scaling functions and wavelets are formed by tensor products of the 1-dimensional scaling functions and wavelets. Consider the 2- dimensional case with $\mathbf{x} = (x_1, x_2)^T$. Denoting the 1-D scaling function by $\phi(x)$ and the 1-D wavelet by $\psi(x)$ one can show that the 2-dimensional scaling function is given by :

$\phi (x_1, x_2) = \phi(x_1)\, \phi(x_2)$.

Similarly, the corresponding wavelet functions are given by :

$\psi^1(x_1, x_2) = \phi(x_1)\, \psi(x_2)$

$\psi^2(x_1, x_2) = \psi(x_1)\, \phi(x_2)$

$\psi^3(x_1, x_2) = \psi(x_1)\, \psi(x_2)$

For an accurate approximation, all the four basis functions must be used at each hidden node. Kugarajah and Zhang have shown that, under certain conditions, a radial basis scaling function $\phi(\| x - x_i \|)$ and wavelet $\psi(\| x\text{-}x_i \|)$ Constitute frame, and that these functions can be used in place of the entire N-dimensional basis, resulting in a savings in storage and execution time while minimally affecting the accuracy of the approximation. The operation of WBFNN is summarized in the following steps :

Step 1. Basis Function Selection: A significant issue in wavelet basis function neural networks is the selection of the basis functions. The wavelet family used in the WBFNN depends on the form of the function F that must be reconstructed. Even though this function is usually unknown, some important details may be obtained by inspecting the problem at hand. For instance, classification usually calls for a discontinuous or quantized function F where all the input data is to be mapped onto one of a few classes. In such cases, discontinuous wavelets, may be used. Continuous wavelets may be used to approximate smoother functions.

Step 2. Center Selection: The location and number of basis functions are important since they determine the architecture of the neural network. Centers at the first (or coarsest) resolution are selected by using the K-means algorithm. Each center at successive resolutions is computed as the mean of two centers at a lower resolution.

Step 3. Training : Training the network involves determining the expansion coefficients associated with each resolution level. These coefficients are determined by using a matrix inversion operation, similar to the operation performed in RBFNN . The centers can also be dynamically varied during the training process till the error in the network prediction falls below a predetermined level. Over-fitting by the network can be avoided by pruning the centers one by one until the network performs at an acceptable level on a blind test database. In this study however, no optimization is performed after center selection.

Step 4. Generalization: In this step, the trained WBFNN is used to predict the output for a new test signal using (3).

4. Applications

4.1. FFNN Results Using RBFNN

The RBFNN was tested using a first order Bessel function $J_1(t)$:

$$t^2 y'' + t y' + (t^2 - 1) y = 0 \quad , t \in [0, 20]$$

4.2. FFNN Results Using WBFNN

Two resolution levels, with 10 centers at the coarsest resolution were selected using the K-Means clustering algorithm. No optimization was performed after center selection to reduce the number of basis functions used. The scaling function used was a Gaussian function : $\phi\ (x ,c) = \exp(-\|x- c\|/ 2\sigma^{2})$ where **c** and σ are the center and spread of the scaling function, respectively. The wavelet functions :

$$\psi\ (x ,c) = ((1-\|x- c\|^2\, 2^m) / 2\sigma^2) \exp(-\|x-c\|^2 2^m / 2\sigma^2) .$$

Where **c** and σ are the center and spread of the wavelet function respectively m is a parameter controlling the dilation of the wavelet, whose value depends on the resolution level. Figure 2 shows the performance of the WBFNN as a forward model

Comparing the results in Figures1and 2, we see that the WBFNN is a better forward model than the RBFNN and the error surface can be illustrated by figure **3.**

4.3. Comparing of performance RBFNN and WBFNN.

FFNN was tested using van der pol equation: $x'' + (x^2 - 1)\, x' + x = 0$, $x_1(0) = 1$, $x_2(0) = 1$Use RBFNN and WBFNN with $\sigma = 0.01$

5. CONCLUSIONS AND FUTURE WORK

This study proposed the use of ANN based forward models in iterative algorithms used for solving multivariate function approximation problems. Two different types of neural networks RBFNN and WBFNN used to represent the forward model. These forward models were used, in iterative scheme, or in combination with an inverse model in feedback configuration, to solve the inverse problem. This type of FFNN consists offers several advantages over numerical models in terms of both implementation of gradient calculations in the updates of the parameters and overall computational cost. One drawback of these approaches is that the forward models are not accurate when the input signals are not similar to those used in the training database. From all above study the results of applying the FFNN to one- and two-dimensional problems were presented. Also we study the comparison between RBFNN and WBFNN to obtain better results. In general, our numerical result shows that it is difficult to RBFNN specify which algorithms will converge faster.

1- For the large problem we treat, we recommend that: try first the RBFNN.

2- Our numerical results shows that, in approximation of the numerical solution of ODE or PDE using RBFNN gives better accuracy than using RBFNN for small dimensional problems.

3- The approximation of function offers the alternative approach of adopting a general purpose optimization method to solve the relevant non - linear approximation problem in feed forward propagation procedures.

4- For high-dimensional problems, RBFNN are potentially valuable, since they may be based on a limited number of centers, which do not have to be placed on a grid throughout the domain .

Another issue that needs to be examined in future work is related with the sampling of the grid points that are used for training. In the above experiments the grid was constructed in a simple way by considering equidistant points. It is expected that better results will be obtained in the case where the grid density will vary during training according to the corresponding error values. This means that it is possible to consider more training points at regions where the error values are higher.

Developing an upper bound for the error $\|\psi_t(x) - \psi_a(x)\| \neq O(m^\alpha)$, require further study, where m is the number of basis function and α is the dimension of the domain.

As the dimensionality increases, the number of training points becomes large. This fact becomes a serious problem for methods that consider local functions around each grid point since the required number of parameters becomes excessively large and, therefore, both memory and computation time requirements become extremely high, and require further study.

References

1. Atalla, .J., 1996 Model Updating Using Neural Networks, Ph.D. Thesis, Virginia Polytechnic Institute and State University,.

2. Cheney, .E. W. and Light, W., 2000. Course in approximation theory, Pub. Books, Colepub. Company,.

3. AL-Mosawi, A. K., 2009 On Training of Feed forward Neural Networks For Approximation Problems , Msc Thesis , Department of Mathematics , College of Education-Ibn Al-Haitham, University of Baghdad , .

4. Yegnanarayana, B., 2000Artificial Neural Networks, Newdelhi ,

5. Tawfiq ,M. ,and Eqhaar ,Q.H. , On Radial Basis Function Neural Networks , Al-Qadisya Journal for Science ,. 12:, No. 3 , 2007 .

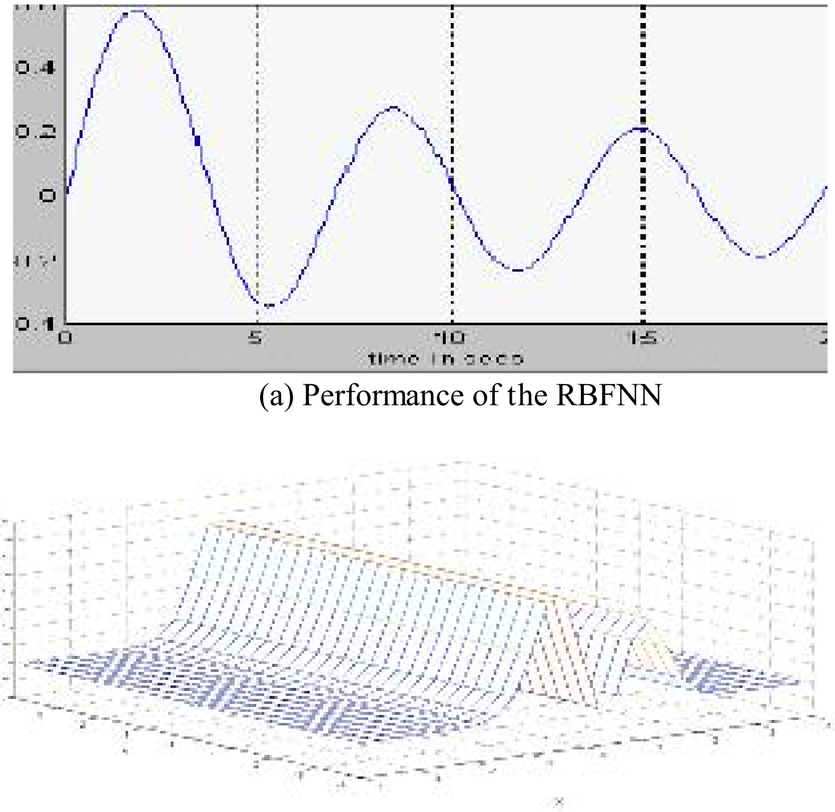
6. Wajdi, B., Chokri, B. A. and Alimi, A. M. Comparison between Beta Wavelets Neural Networks, RBF Neural Networks and Polynomial Approximation for 1D, 2D Functions Approximation , ROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TEC-

HNOLOGY, 13:, 2006 , ISSN 1307- 6884.

7. Bakshi, B.R. and Stephanopoulos, G. " Wave net: A multiresolution neural network with localized learning ", AIChEJ.,.39, No.1, pp. 57- 81, 1993.

8. Mallat, S.G. " A theory for multiresolution signal decomposition: The wavelet representation" IEEE Trans. Pottern Analysis Mach. Intelligence,.11:, No.7, pp.674 - 693, 1989.

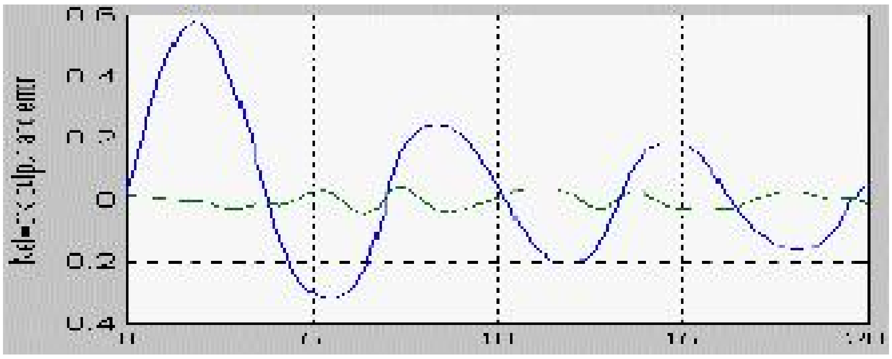
9. Kugarajah, T. and Zhang, Q. "Multidimensional wavelet frames", IEEE Transactions on Neural Network s.6:, No.6, pp.1552-1556, 1995.



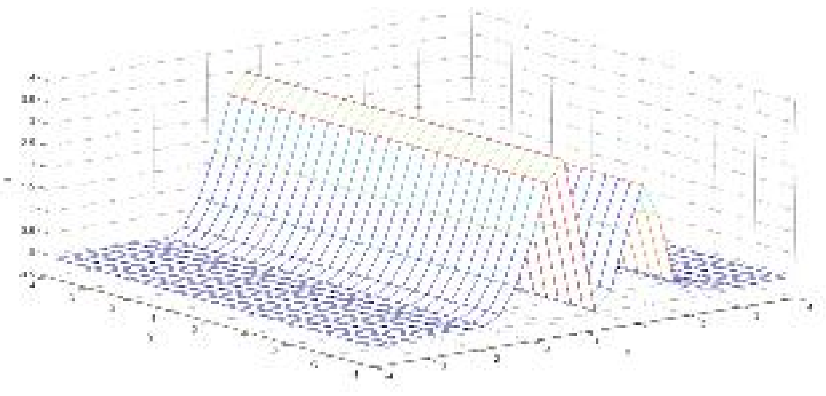
(a) Performance of the RBFNN

(b) Results of iterative RBFNN

Figure (1) : RBFNN forward model .



(a) performance of the WBFNN .



(b) Results of iterative WBFNN .

Fig. (2) : WBFNN forward model

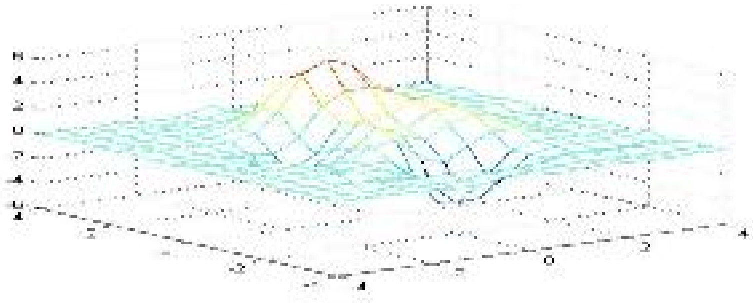
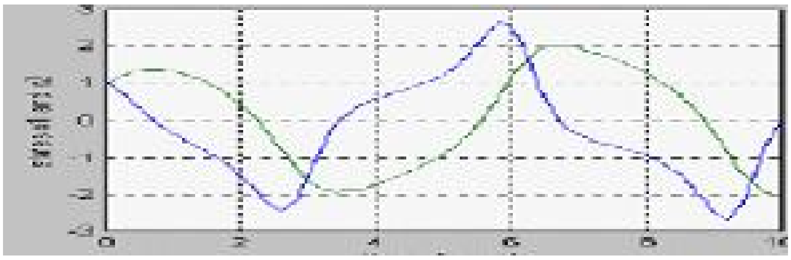
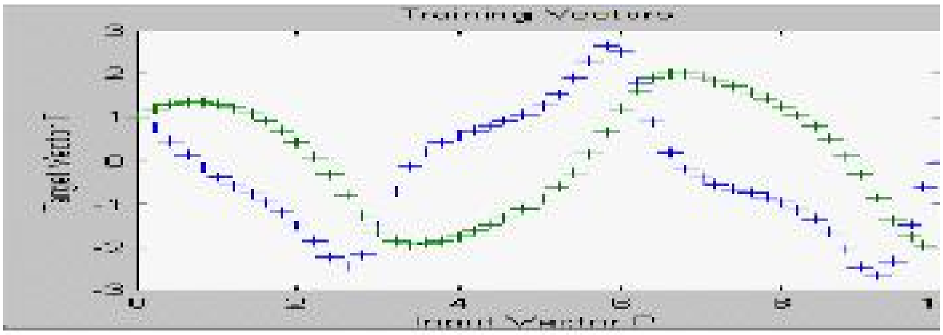


Fig.e 3. compare between RBFNN and WBFNN from figure (1) & (2)



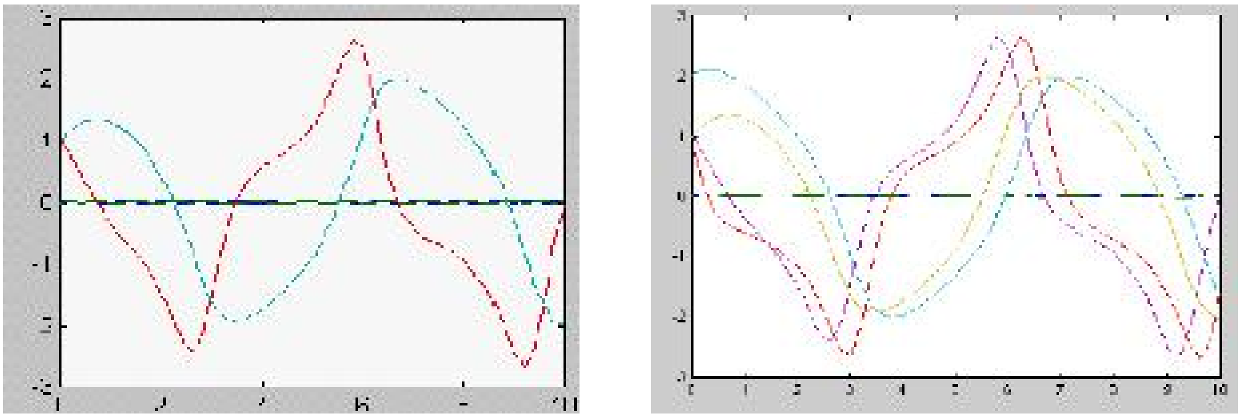
(a)



(b)

Fig. 4. performance of RBFNN: (a) result (b) error

Figure 5. Result and error of WBFNN .



(a) (b)

Figure 6. compare between RBFNN and WBFNN (a) results (b) error

مجلة ابن الهيثم للعلوم الصرفة والتطبيقية المجلد23 (2) 2010

حول المقارنة بين الشبكات العصبية ذي دوال الأساس الشعاعية و دوال الأساس المتذبذبة

لمى ناجي محمد توفيق و تغريد عبد المجيد رشيد

قسم الرياضيات ،كلية التربية ابن الهيثم ، جامعة بغداد

الخلاصه

يتضمن هذا البحث دراسة وتصميم شبكتان عصبيتين صناعيتين ذاي التغذية التقدمية . الأول يستخدم RBFNN والثاني يستخدم WBFNN لتقريب تطبيق من فضاء المدخلات إلى فضاء المخرجات ثم دربنا الشبكات باستخدام خوارزمية التدريب المرتد من النوع C G للحصول على المخرجات المطلوبة ثم طبقنا تلك الشبكات على بعض الأمثلة لحل معادلات تفاضلية ومن ثم عرضنا نتائج تلك التطبيقات .